Tivoli. software

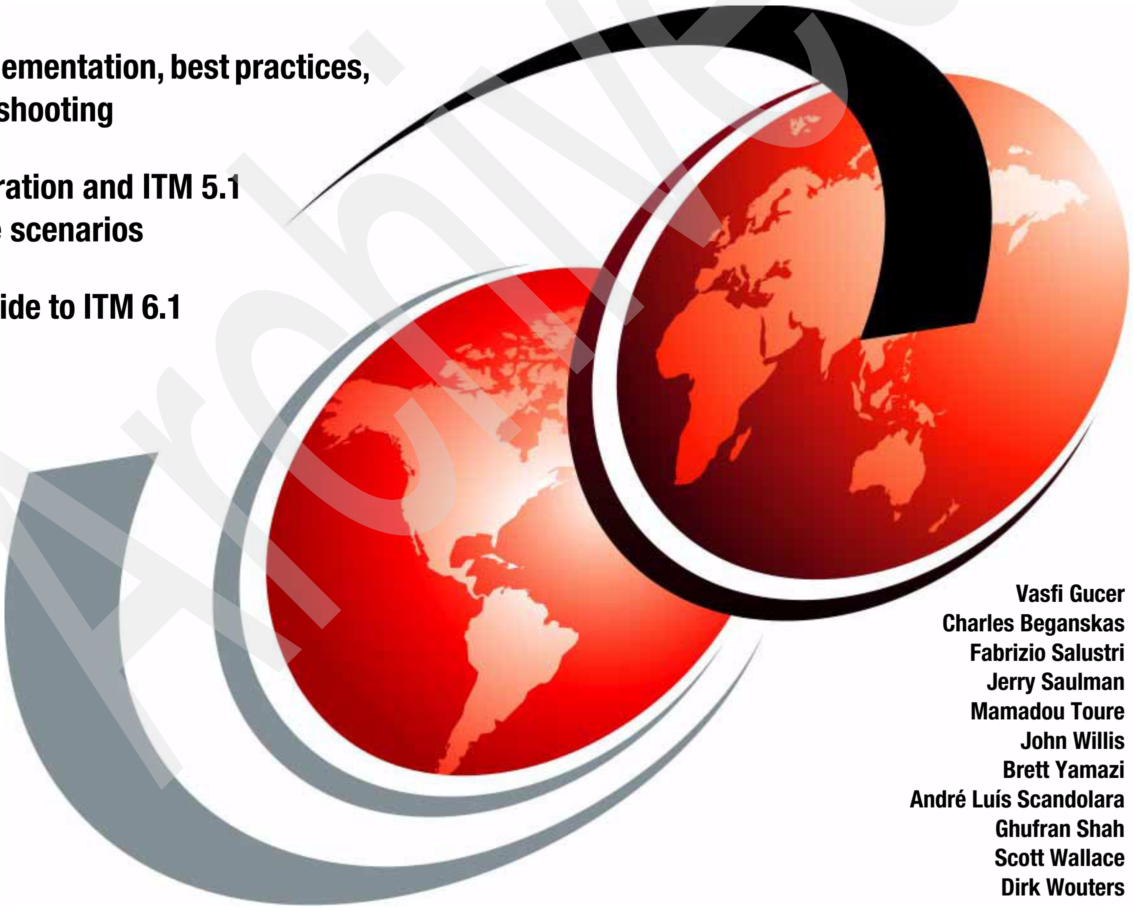
IBM

Getting Started with IBM Tivoli Monitoring 6.1 on Distributed Environments

ITM 6.1 implementation, best practices, and troubleshooting

DM 3.7 migration and ITM 5.1 coexistence scenarios

Insider's guide to ITM 6.1



Vasfi Gucer
Charles Beganskas
Fabrizio Salustri
Jerry Saulman
Mamadou Toure
John Willis
Brett Yamazi
André Luís Scandolara
Ghufran Shah
Scott Wallace
Dirk Wouters

Redbooks

ibm.com/redbooks

International Technical Support Organization

Getting Started with IBM Tivoli Monitoring 6.1 on Distributed Environments

December 2005

SG24-7143-00

Note: Before using this information and the product it supports, read the information in "Notices" on page xxvii.

First Edition (December 2005)

This edition applies to IBM Tivoli Monitoring Version 6, Release 1.

Contents

Figures

Tables

Examples

Notices

Trademarks

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

AIX®
AS/400®
Candle®
Candle Management Server®
Candle Management Workstation®
CandleNet®
CandleNet Portal®
CICS®
DB2®
Domino®
HACMP™
i5/OS®
IBM®
Informix®
iSeries™
Lotus®
MQSeries®
NetView®
OMEGAMON®
OMEGAMON Monitoring Agent®
OS/400®
Passport Advantage®
RACF®
Redbooks™
Redbooks (logo) ™
Tivoli Enterprise™
Tivoli Enterprise Console®
Tivoli Management Environment®
Tivoli®
TME®
WebSphere®
z/OS®
zSeries®

The following terms are trademarks of other companies:

iPlanet, Java, JavaHelp, JDBC, Solaris, Sun, Sun Java, SunOS, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

BizTalk, Microsoft, SharePoint, Windows server, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Xeon, 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

The IBM® Tivoli® Monitoring Version 6.1 solution is the next generation of the IBM Tivoli family of products that help monitor and manage critical hardware and software in distributed environments. IBM Tivoli Monitoring 6.1 has emerged from the best of the IBM Tivoli Monitoring V5 and OMEGAMON® technologies. Integration of these products creates a unique and comprehensive solution to monitor and manage both z/OS® and distributed environments.

IBM Tivoli Monitoring 6.1 is easily customizable and provides real-time and historical data that enables you to quickly diagnose and solve issues with the new GUI via the IBM Tivoli Enterprise™ Portal component. This common, flexible, and easy-to-use browser interface helps users to quickly isolate and resolve potential performance problems.

This IBM Redbook covers planning, architecture, tuning, implementation, and troubleshooting of IBM Tivoli Monitoring 6.1. In addition, we offer scenarios for migration from Distributed Monitoring 3.7, and IBM Tivoli Monitoring 5.X coexistence with IBM Tivoli Monitoring 6.1.

This book is targeted for IT specialists who will be working on new IBM Tivoli Monitoring 6.1 installations or implementing migration from Distributed Monitoring V3.7 or IBM Tivoli Monitoring 5.X coexistence.

The team that wrote this redbook

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

Figure 1 Part of the residency team, from left : Charles, Fabrizio, Vasfi, Brett, Mamadou, John, Jerry

Vasfi Gucer is an IBM Certified Consultant IT Specialist at the ITSO Austin Center. He was with IBM Turkey for 10 years, and has worked at the ITSO since January 1999. He has more than 13 years of experience in teaching and implementing systems management, networking hardware, and distributed platform software. He has worked on various Tivoli customer projects as a Systems Architect and Consultant. Vasfi is also a Certified Tivoli Consultant.

Charles Beganskas is an IBM IT Specialist working in IBM Global Services Division, Poughkeepsie, NY for seven years. His skills include IBM Tivoli Monitoring V5.1.2, Distributed Monitoring V3.7, and Tivoli Framework V4.1.1, with expertise in mySAP monitoring and actively developing best practices for monitoring mySAP systems. He holds a RedHat Certified Engineer certification and has extensive experience in UNIX®/Linux® administration and system management. He has supported IBM Tivoli Monitoring for the IBM Global Account for three years.

Fabrizio Salustri is a Software Support Specialist working for Italy IMT in Tivoli Customer Support within IBM Global Services. He worked for IBM since 1996, and has extensive experience with Tivoli products. Throughout his career, Fabrizio has been involved in several projects implementing Tivoli solutions for important clients of IBM Italy. Before joining the Tivoli Support team, he worked as a Certified AIX® System Administrator in AIX Technical Support. In March 2005. he got an IBM Tivoli Monitoring 5.1.1 Deployment Professional Certification.

Jerry Saulman, a member of the IBM Tivoli Software Global Response Team, has 10 years of experience with Tivoli products including as a customer doing a global (64 countries) deployment of enterprise products. For the past seven years he has presented at Tivoli technical conferences, user group meetings, SHARE conferences, and other technical events. He is the author of numerous Tivoli Field Guides about business and technical issues. He helped to design and support the IBM Tivoli implementation at the 2000 Olympics.

Mamadou Toure is an IT Specialist, IBM Tivoli Certified Consultant working as senior consultant for Groupe CGI Inc in Montreal. He holds a Masters degree in Computer Science from the University of Paraiba. He has more than 10 years of experience in the IT sector and has been involved in successful design and implementation of enterprise system management tools such as NetView®, HP - Openview, E-Health, and Tivoli suite at various customer environments. Mamadou has extensive experience in designing, implementing, and supporting such Tivoli products as Framework 4.x, DM3.x, ITM 5.1.x, Tivoli Enterprise Console 3.x, TCM 4.x, and NetView.

John Willis is a Tivoli Enterprise Certified Instructor, Consultant, and Lead Architect for Gulf Breeze Software (gulfsoft.com). He has more than 20 years in IT with five years of experience working with Tivoli. John is a frequent speaker at SHARE.org on DM, Tivoli Enterprise Console, and the Workbench. His current area of expertise includes working with CIM and WMI.

Brett Yamazi is a Senior Software Engineer with IBM/Tivoli Software. He has worked with IBM for eight years assisting customers and IBM engineers in support centers worldwide with software in the Availability and Business Service Management areas, with focus on Distributed Monitoring, IBM Tivoli Monitoring, Tivoli Decision Support, and Tivoli Data Warehouse. He graduated with degrees in Physics and Philosophy from the University of Texas at Austin, and received an MBA from St. Edward's University.

André Luís Scandolara is an IT Specialist working for IGS Strategic Outsourcing/SDC Brazil on the Tivoli Support Team providing deployment and support in Tivoli systems for outsourcing customers in Brazil. He got his degree as Bachelor (BSc) in Systems Analysis (2001) from Pontificial Catholic University of Campinas, Brazil, and post-graduate in Network Computers (2003) from State University of Campinas, Brazil.

Ghufran Shah is a Tivoli Certified Enterprise Consultant and Instructor with os-security.com in the United Kingdom. He has eight years of experience in Systems Development and Enterprise Systems Management. He holds a degree in Computer Science from the University of Bradford. His areas of expertise include Tivoli Systems Management Architecture, Implementation, and Tivoli Training, together with Business Process Improvement and Return on Investment modeling. He has written extensively about event Management, Monitoring, and

Business Systems Management integration and has taught Tivoli courses worldwide.

Scott Wallace is a Senior Software Engineer working in IBM Tivoli Customer Support at Research Triangle Park, North Carolina. He joined IBM in 1995 and has been with IBM Tivoli since 1998. He has more than 13 years of experience in systems management and distributed platform software. Scott is an IBM Tivoli Certified Professional and is certified in ITIL.

Dirk Wouters is a member of the EMEA Global Response Team for IBM Tivoli Monitoring. He worked with Candle® and OMEGAMON for more than 15 years. Dirk has expertise on the internal architecture of the OMEGAMON (both on z/OS and distributed) and most of the monitoring agents. He is based in Belgium.

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

Budi Darmawan, Betsy Thaggard
International Technical Support Organization, Austin Center

Jon Austin, Daniel Biskar, Rod Bowser, Jim Carey, Catherine Cook, John Cook, Adrian Edwards, Jennifer Egan, Clyde Foster, Bowman Hall, Greg Hamm, Maureen Hannah, Daniel Hernandez, Axel Hoppe, Thad Jennings, Brent Kellis, Mike Mallo, Francisco Moraes, Ed Moreno, Stefan Muller, Heath Newburn, Jan Love, Kam Lee, Liz Poggemeyer, Joe Saunders, Urs Schwarzenbach, Glenn Spiegelman, Ben Stern, Carl Swanson, Tod Thorpe, Divyesh Vaidya, Lorraine Vassberg, Troy Volin, Brooks York, Stephan Woehl
IBM USA

Ingo Averdunk
IBM Germany

Francesco Lupini, Alessandra Masci
IBM Italy

Dennis A Perpetua
IBM USA

Jason Shamroski
Frank Tate
Gulf Breeze Software (gulfsoft.com)

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners and/or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll 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 Redbooks™ to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

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

  ibm.com/redbooks

- Send your comments in an e-mail to:

  redbook@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. JN9B Building 905
  11501 Burnet Road
  Austin, Texas 78758-3493

1

Introduction

This chapter introduces the concepts and components behind IBM Tivoli Monitoring Version 6.1. If you are new to Version 6.1, you can also refer to *IBM Tivoli Monitoring Version 6.1.0. Quick Start Guide*, SC32-1802, which provides concise information about the product.

Important: This book focuses on the distributed environments. For z/OS environments, there is another redbook (in progress) called *IBM Tivoli OMEGAMON V3.1.0 Deep Dive on z/OS*, SG24-7155. The expected availability date for this new book is February 2006, with a draft version by the end of 2005.

This chapter has the following sections:

- Enterprise management challenges
- IBM Tivoli Monitoring solutions
- What's new for OMEGAMON XE/DE clients?
- What's new for Distributed Monitoring V3.7 clients?
- IBM Tivoli Monitoring 6.1 value proposition
- IBM Tivoli Monitoring 6.1 components
- What is new in IBM Tivoli Data Warehouse V2.1?

1.1 Enterprise management challenges

Many readers will be familiar with the challenges faced by IT departments and IT support teams. The nature of ever-changing business demands and market dynamics often put strains on IT resources, and we are constantly told to “do more, with less.”

The reality we are facing today includes situations where problems occur in systems and the environment well before we are notified, causing IT staff to operate in a reactive, “firefighting” mode. In some environments, the fires seem to get bigger as the pressure, tight deadlines, and high profile of business solutions shadows the need to plan an effective systems management solution.

Traditional enterprise management has to change, as the modus operandi tend to include most if not all of the following:

- It is reactive, not proactive.
- Resources may be healthy while customer service levels are not acceptable.
- Events describe problems, not corrective actions.
- Events flow into the Operations Room at an incredibly high rate, and “event storms” have performance impact on systems.
- Fixes are typically manual and inefficient.
- Cannot prioritize problems because impacts are unknown.
- Cannot detect most problems: More than 50% of all problems are reported through the help desk.
- Organizational boundaries breed incompatible tools, making end-to-end management and integration very difficult.
- Lack of vision and strategic direction increases costs.

1.1.1 Business driving forces

It can be noted that there are some key business driving forces behind an effective enterprise management, one being the need to improve the quality of service delivery and reduce the resources required to implement and use new information technologies. In addition the following factors must be taken into account when planning and defining the vision of enterprise management.

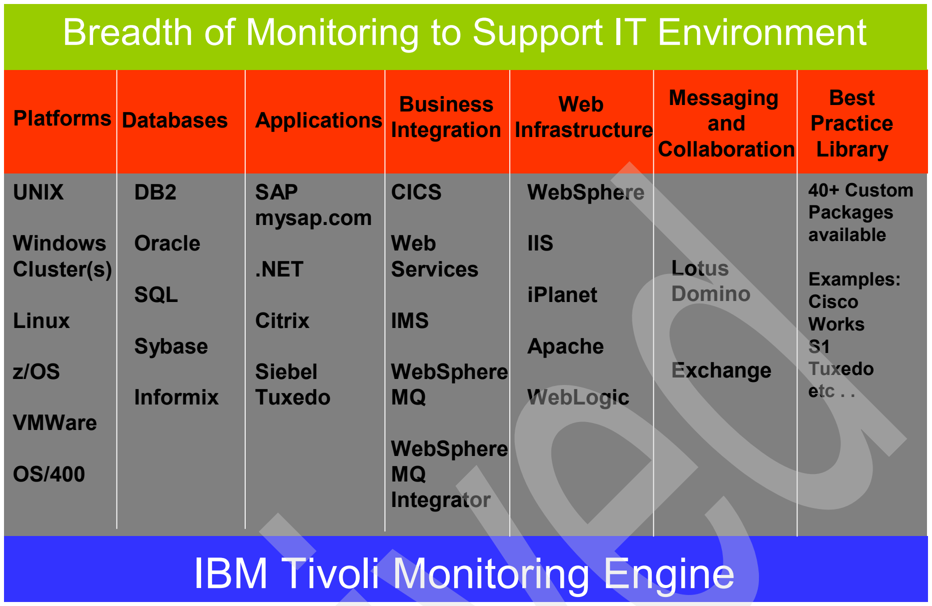
- The need to increase revenues, reduce costs, and compete more effectively.
- Companies need to deploy informational applications rapidly, and provide business users with easy and fast access to business information that reflects the rapidly changing business environment. Enterprise Management solutions must be transparent to the business solutions being delivered, and

they must be proactive to detect, resolve, and escalate potential issues which may impact the business service being delivered.

- The need to manage and model the complexity of today’s business environment.
- Corporate mergers and deregulation means that companies today are providing and supporting a wider range of products and services to a broader and more diverse audience than ever before. Understanding and managing such a complex business environment and maximizing business investment is becoming increasingly more difficult. Enterprise management systems provide more than just basic monitoring mechanisms; they also offer sophisticated issue detection, event correlation, and application and transaction discovery and performance management tools that are designed to handle and process the complex business information associated with today’s business environment.
- The need to reduce IT costs and leverage existing corporate business information.
- The investment in IT systems today is usually a significant percentage of corporate expenses, and there is a need not only to reduce this overhead, but also to gain the maximum business benefits from the information managed by IT systems. New information technologies like corporate intranets, thin-client computing, and subscription-driven information delivery help reduce the cost of deploying business intelligence systems to a wider user audience, especially information consumers like executives and business managers. Maintaining the maximum uptime of these systems is becoming more and more critical.

1.2 IBM Tivoli Monitoring solutions

IBM Tivoli Monitoring solutions provide a means to manage distributed resources through centralized control and configuration, and for many years IBM Tivoli has been a market leader in enterprise monitoring solutions. IBM Tivoli Monitoring has been the backbone for availability monitoring across operating systems and application components.


Breadth of Monitoring to Support IT Environment

Platforms	Databases	Applications	Business Integration	Web Infrastructure	Messaging and Collaboration	Best Practice Library
UNIX	DB2	SAP mysap.com	CICS	WebSphere	Lotus Domino	40+ Custom Packages available
Windows Cluster(s)	Oracle	.NET	Web Services	IIS	Exchange	Examples: Cisco Works S1 Tuxedo etc . .
Linux	SQL	Citrix	IMS	iPlanet		
z/OS	Sybase	Siebel Tuxedo	WebSphere MQ	Apache		
VMWare	Informix		WebSphere MQ Integrator	WebLogic		
OS/400						

IBM Tivoli Monitoring Engine

Figure 1-1 IBM Tivoli Monitoring solutions

IBM Tivoli Monitoring solutions provide a solid foundation for the development of management solutions addressing the complex needs of today's IT infrastructures. A set of modules built on top of IBM Tivoli Monitoring provide a comprehensive set of solutions for companies facing the challenge of monitoring composite application infrastructures. These modules are delivered through a set of offerings that include:

- IBM Tivoli Monitoring for Applications
- IBM Tivoli Monitoring for Business Integration
- IBM Tivoli Monitoring for Databases
- IBM Tivoli Monitoring for Messaging and Collaboration

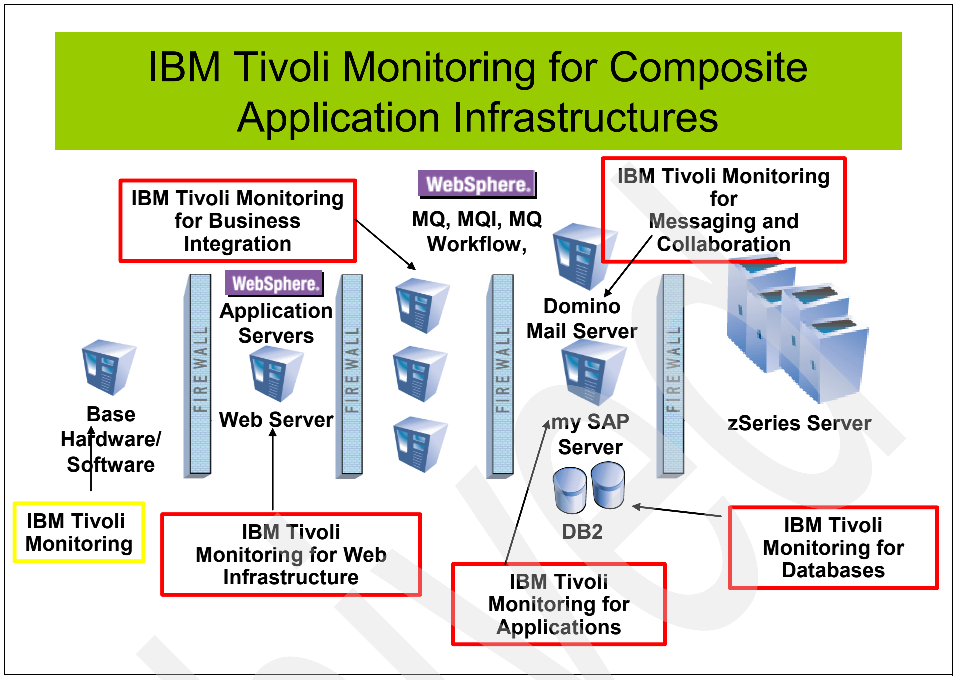

IBM Tivoli Monitoring for Composite Application Infrastructures
IBM Tivoli Monitoring for Business Integration
WebSphere.
MQ, MQI, MQ Workflow,
IBM Tivoli Monitoring for Messaging and Collaboration
WebSphere.
Application Servers
Domino Mail Server
FIRE WALL
Base Hardware/ Software
Web Server
my SAP Server
zSeries Server
DB2
IBM Tivoli Monitoring
IBM Tivoli Monitoring for Web Infrastructure
IBM Tivoli Monitoring for Applications
IBM Tivoli Monitoring for Databases

Figure 1-2 Monitoring composite application infrastructures

The latest generation of IBM Tivoli Monitoring solutions have service-oriented themes, and are now focused on:

- Consolidating monitoring platforms.
- Ensuring customers can take advantage of and realize the return on their monitoring investments.
- Improving visualization and analysis of monitoring data.
- Improving management of monitoring environment.
- Simplifying the installation, configuration, and deployment of the solutions with a simplified user interface.
- Improving the integration of Tivoli products.
- Elevating the value of products through process integration.

1.3 What's new for IBM Tivoli Monitoring 5.x clients?

IBM Tivoli Monitoring 6.1 maintains many of the components of IBM Tivoli Monitoring 5.x, such as integration with the Common Information Model (CIM) and Windows® Management Instrumentation (WMI), and the powerful and extremely customizable resource model based monitoring. Highlights of IBM Tivoli Monitoring 6.1 are:

- Clients can adopt some or all IBM Tivoli Monitoring 6.1 components.
- The new Integrated Portal (Tivoli Enterprise Portal) provides the user interface for IBM Tivoli Monitoring 6.1, and integrating applications replaces the Web Health Console and provides a single management and operational view into the enterprise infrastructure. This portal provides tremendous value and is integral to the enterprise monitoring solution. Some of its features are:
  - Improvement of usability and visualization
  - Integrated Administration and Runtime Portal
  - Authorization control for systems, applications, functions, and geographical views
  - Views that can be customized and linked together for efficient use by a specific audience
  - Easy administration of management rules
- All IBM Tivoli Monitoring 5.1 configurations remain supported.
- Ability to integrate data coming from various sources such as IBM Tivoli Monitoring 5, OMEGAMON, OMEGAMON Z, IBM Tivoli Enterprise Console®, and IBM Tivoli Data Warehouse.
- Enhanced process control with a single console for incident and problem management supporting custom workflow (through linking and custom workspace).
- Deep performance-based data and granular information.
- Granular historical data providing.
  - Trending with the Data Warehouse.
  - Forensic data for problem reconstruction.
  - Simplified schema for easier manipulation and extraction of data.
- Web Services (SOAP) interface to access real time and historical information.

IBM Tivoli Monitoring 5.x and IBM Tivoli Monitoring 6.1 coexistence is covered in detail in Chapter 6, "Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint" on page 317.

1.4 What's new for OMEGAMON XE/DE clients?

OMEGAMON clients gain many enhancements and new features such as:

- Management of additional resources, such as Citrix and .NET support
- More agents and more agent deployment support
- Deployment support in the following areas:
  - Centralized agent deployment capabilities
  - Enhanced internal infrastructure management
  - Integration with Tivoli Management Framework environment
- Command line interface support and Web Services interface
- Increased scalability
- Enhanced integration with Tivoli products such as:
  - IBM Tivoli Enterprise Console
  - IBM Tivoli Monitoring 5.1
- Enhanced Data Warehousing such as Aggregation and Pruning controls to manage the Tivoli Data Warehouse, which replaces the Candle Data Warehouse

Refer to Chapter 7, "OMEGAMON XE Upgrade" on page 407 for details about migration from OMEGAMON XE to IBM Tivoli Monitoring 6.1.

1.5 What's new for Distributed Monitoring V3.7 clients?

Customers who still use Distributing Monitoring 3.7 now have the opportunity to upgrade to a modern Monitoring console, warehouse, and monitoring infrastructure.

The upgrade procedure provides these benefits:

- Automated programmatic upgrade to IBM Tivoli Monitoring 6.1
- The new visualization interfaces, ease of use, and the new Data Warehouse
- The IBM Supported upgrade procedures to bring your modules forward

Distributing Monitoring 3.7 migration is covered in detail in Chapter 5, "Tivoli Distributed Monitoring V3.7 upgrade" on page 239.

1.6 IBM Tivoli Monitoring 6.1 value proposition

IBM Tivoli Monitoring 6.1 continues the paradigm of enterprise resource monitoring, and this release has several important themes, as shown in Figure 1-3.

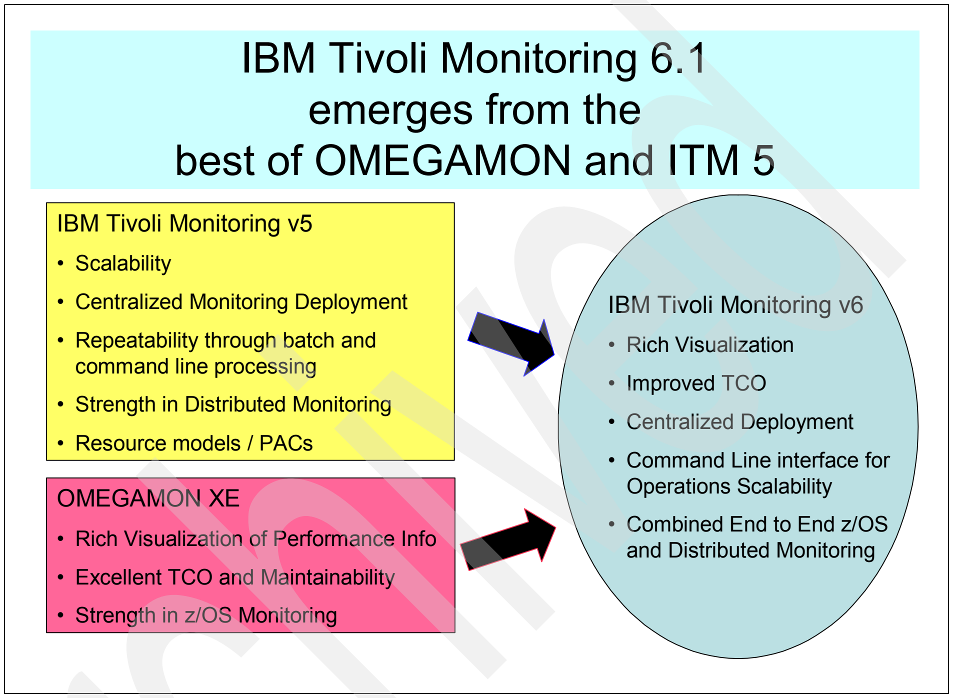

IBM Tivoli Monitoring 6.1
emerges from the
best of OMEGAMON and ITM 5
IBM Tivoli Monitoring v5
• Scalability
• Centralized Monitoring Deployment
• Repeatability through batch and command line processing
• Strength in Distributed Monitoring
• Resource models / PACs
OMEGAMON XE
• Rich Visualization of Performance Info
• Excellent TCO and Maintainability
• Strength in z/OS Monitoring
IBM Tivoli Monitoring v6
• Rich Visualization
• Improved TCO
• Centralized Deployment
• Command Line interface for Operations Scalability
• Combined End to End z/OS and Distributed Monitoring

Figure 1-3 IBM Tivoli Monitoring 6.1: the best of OMEGAMON and ITM 5

The combination of two best-of-breed solutions—OMEGAMON and IBM Tivoli Monitoring 5.x—has resulted in IBM Tivoli Monitoring 6.1. The key themes in this release are namely:

- An easy-to-use solution:
  - Flexible interface for personalizing information
  - Integration of event management, real-time and historical analysis
- Visualization of information:
  - Robust real-time and historical reports
  - Focus on availability and performance
  - Monitoring infrastructure visibility

- Reduce Total Cost of Ownership:
  - Simple agent and management rule deployment
  - Lightweight agents
  - Support for new technology (clusters, virtualization)
- Integration of products and processes:
  - Enterprise view for Distributed and zSeries®.
  - Integrated view of existing monitoring, new monitoring, and IBM Tivoli Enterprise Console.
  - Added focus of the IT Infrastructure Library, (ITIL), which is a series of documents that are used to aid the implementation of a framework for IT Service Management (ITSM). This framework defines how Service Management is applied within specific organizations, and is completely customizable for applications within any type of business or organization that has a reliance on IT infrastructure.

1.7 IBM Tivoli Monitoring 6.1 components

IBM Tivoli Monitoring 6.1 has been designed to provide access to information that is crucial to daily operations. This information related to the availability and performance for components, applications, and services within your enterprise infrastructure.

We now introduce the various components that provide the technology for this information.

Tivoli Monitoring Services

Tivoli Monitoring Services is the framework for IBM Tivoli Monitoring 6.1 and comprises all components as well as describes how they interact. Those components include but are not limited to:

- Tivoli Enterprise Monitoring Agent (TEMA)
- Tivoli Enterprise Monitoring Server (TEMS)
- Tivoli Enterprise Portal Server (TEPS)
- Tivoli Enterprise Portal (TEP)

Figure 1-4 on page 10 shows the IBM Tivoli Monitoring components.

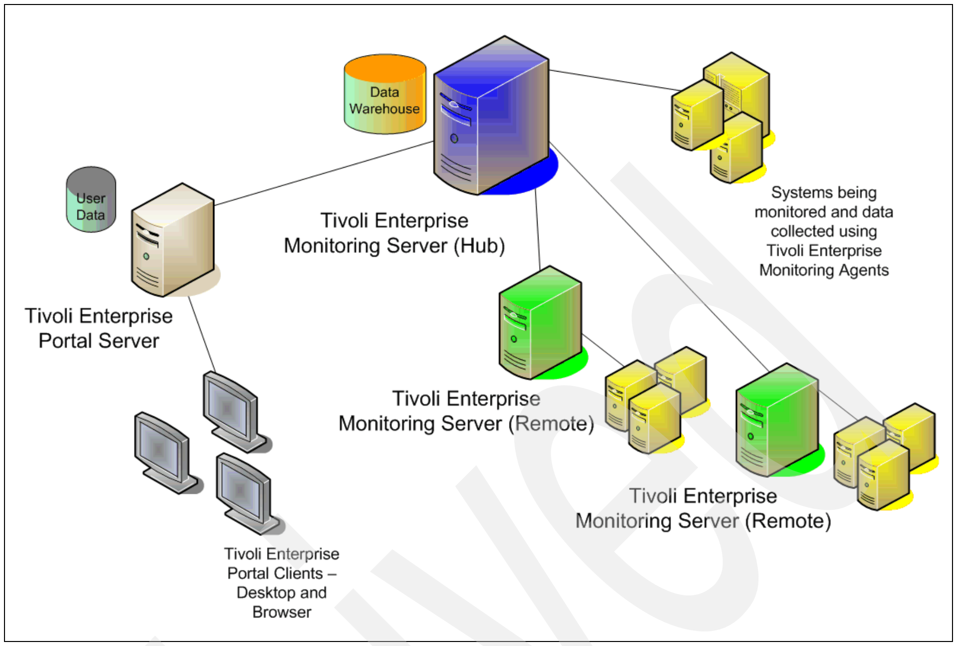

Data Warehouse
User Data
Tivoli Enterprise Monitoring Server (Hub)
Systems being monitored and data collected using Tivoli Enterprise Monitoring Agents
Tivoli Enterprise Portal Server
Tivoli Enterprise Monitoring Server (Remote)
Tivoli Enterprise Monitoring Server (Remote)
Tivoli Enterprise Portal Clients – Desktop and Browser

Figure 1-4 BM Tivoli Monitoring 6.1 components

Tivoli Enterprise Monitoring Agent

The Tivoli Enterprise Monitoring Agents (TEMA) are data collectors within your monitoring solution, and are installed to gather data from one or more systems that you need to monitor. The data that is collected is sent to a central repository, called Tivoli Data Warehouse. TEMAs collects information about the attributes of a particular managed system. Examples of agents are:

- Operating System Agent.
- Universal Agent: The Tivoli Universal Agent is a monitoring agent you can configure to monitor any data you collect. It enables you to integrate data from virtually any platform and any source, such as custom applications, databases, systems, and subsystems.
- Application Agents: These collect data from databases, WebSphere® Application Server, WebSphere MQ, and so forth.

At General Availability (GA) time, the following Operating System Agents will be supported:

- AIX 5.1, 5.2, 5.3
- Solaris™ 8, 9, 10
- HP UX 11i, 11.23
- Windows 2000 Server, Advanced Server

- Windows XP Pro
- Windows 2003 Server 32-bit (SE & EE)
- OS/400® 5.2, 5.3
- RHEL 2.1
- RHEL 3 (Intel®, zSeries)
- RHEL 4 (Intel, AMD64/EM46T, zSeries)
- SLES 8 (Intel, zSeries)
- SLES 9 (Intel, zSeries)

Note: Windows 2003 Server 64-bit will be supported post-GA as a fix pack.

Tivoli Enterprise Monitoring Server

The Tivoli Enterprise Monitoring Server (TEMS) is the central repository of data that comes from the TivolI Enterprise Monitoring Agents.

It stores the definitions for conditions that indicate a problem with a particular resource and controls the security for your monitoring solution.

Each enterprise monitoring solution must contain one Hub TEMS and can include multiple remote TEMSs, which are used to provide scalability in large installations.

At GA time, the following platforms will be supported for TEMS:

- AIX 5.2, 5.3
- Solaris 9, 10
- Windows 2000 Server, Advanced Server
- Windows XP Pro (demo only)
- Windows 2003 Server 32-bit (SE & EE)
- RHEL 4 (Intel, zSeries)
- SLES 9 (Intel, zSeries)

Tivoli Enterprise Portal Server

The Tivoli Enterprise Portal Server (TEP) functions as a repository for all user data, such as the user IDs and user access control for the monitoring data, meaning what data each user will be able to access and how it is displayed.

The TEPS connects to the Hub TEMS and provides a consistent look and feel for the users. The database to store TEP server information can be:

- MS SQL Server 2000
- IBM UDB 8.1
- IBM UDB 8.2

At GA time, the following platforms will be supported for TEPS:

- Windows 2000 Server, Advanced Server
- Windows XP Pro (supported for demo purposes only)
- Windows 2003 Server 32-bit (SE & EE)
- RHEL 4 (Intel, zSeries)
- SLES 9 (Intel, zSeries)

Tivoli Enterprise Portal

The Tivoli Enterprise Portal (TEP) is the consolidated user interface for IBM Tivoli Monitoring and is used to connect to the Tivoli Enterprise Portal Server.

The Tivoli Enterprise Portal can be launched from an Internet Explorer browser, or can be installed as a client application on a workstation.

Important: Internet Explorer is the only browser supported for TEP.

At GA time, the following platforms will be supported for TEP:

- Windows 2000 Pro
- Windows 2000 Server, Advanced Server
- Windows XP Pro
- Windows 2003 Server 32-bit (SE & EE)
- RHEL 4 (Intel)
- SLES 9 (Intel)

Figure 1-5 on page 13 shows the Tivoli Enterprise Portal (browser mode).

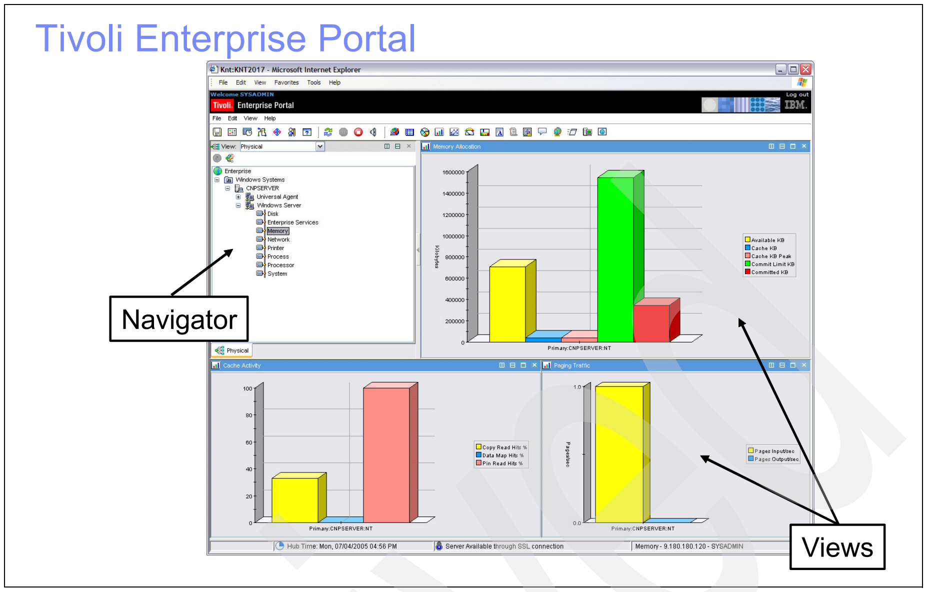


Figure 1-5 Tivoli Enterprise Portal

1.8 What is new in IBM Tivoli Data Warehouse V2.1?

This is a summary of the features provided in IBM Tivoli Data Warehouse V2.1:

- Step 1 of the Extract, Transform, and Load (ETL1) procedure is no longer required.
- The data is no longer aggregated before it is loaded into the data warehouse.
- Crystal Enterprise is no longer needed.
- There are no more data marts.
- The various data collectors have a common core and application-specific plugins.
- Historical and-real time data can aggregate/prune, cross-correlate, and hold process/business views.
- The Tivoli Enterprise Portal can display real-time and historical data and various chart forms.

In summary, the Tivoli Data Warehouse is more useful than ever before.

Refer to Chapter 4, “Historical summarized data” on page 185 for more information about Tivoli Data Warehouse V2.1.

2

Architecture and planning

This chapter explains the IBM Tivoli Monitoring architecture and how each component operates within an IBM Tivoli Monitoring installation. We explore four architectural designs for IBM Tivoli Monitoring using scenarios based on several factors: number of agents, hardware availability, and network restrictions. In addition, an overview section covers IBM Tivoli Monitoring agent deployment using several unique strategies.

This chapter discusses the following:

- IBM Tivoli Monitoring components
- IBM Tivoli Monitoring deployment scenarios
- Scalability
- Agent deployment architecture

2.1 IBM Tivoli Monitoring components

An IBM Tivoli Monitoring installation consists of various components collectively labeled the Tivoli Monitoring Services framework. This framework is a combination of several vital components. Additionally, optional components can be installed which extend the monitoring functionality of this framework. For platform support details for all the major IBM Tivoli Monitoring components, refer to “Platform support matrix for IBM Tivoli Monitoring” on page 22.

Figure 2-1 shows the IBM Tivoli Monitoring component model.

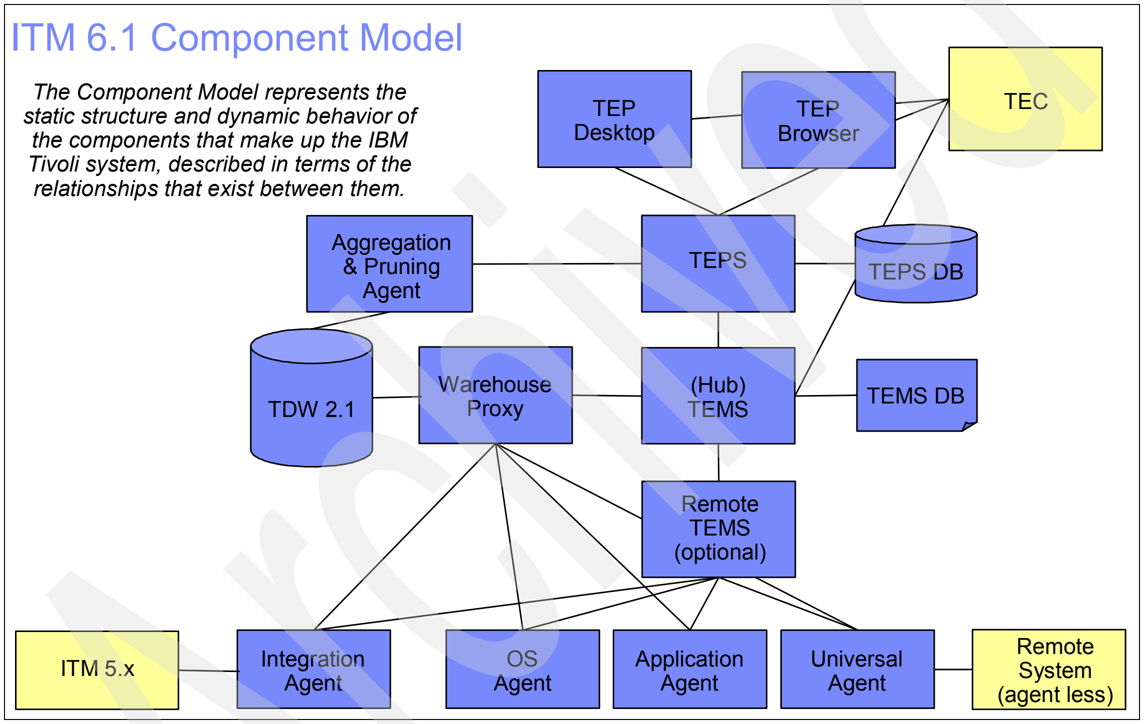

ITM 6.1 Component Model
The Component Model represents the static structure and dynamic behavior of the components that make up the IBM Tivoli system, described in terms of the relationships that exist between them.
TEP Desktop
TEP Browser
TEC
Aggregation & Pruning Agent
TEPS
TEPS DB
TDW 2.1
Warehouse Proxy
(Hub) TEMS
TEMS DB
Remote TEMS (optional)
ITM 5.x
Integration Agent
OS Agent
Application Agent
Universal Agent
Remote System (agent less)

Figure 2-1 IBM Tivoli Monitoring 6.1 component model

Every IBM Tivoli Monitoring installation requires the following components:

- Tivoli Enterprise Monitoring Server (TEMS)

  The Tivoli Enterprise Monitoring Server (referred to as the *monitoring server*) is the initial component to install to begin building the IBM Tivoli Monitoring Services foundation. It is the key component on which all other architectural components depend directly. The TEMS acts as a collection and control point for alerts received from agents, and collects their performance and availability data.

The TEMS is responsible for tracking the heartbeat request interval for all Tivoli Enterprise Management Agents connected to it.

The TEMS stores, initiates, and tracks all situations and policies, and is the central repository for storing all active conditions and short-term data on every Tivoli Enterprise Management Agent. Additionally, it is responsible for initiating and tracking all generated actions that invoke a script or program on the Tivoli Enterprise Management Agent.

The TEMS storage repository is a proprietary database format (referred to as the *Enterprise Information Base - EIB*) grouped as a collection of files located on the Tivoli Enterprise Monitoring Server.

These files start with a filename prefix `qa1` and are located in:

– <installation_dir/tables>/<tems_name>
– <installation_dir>: IBM Tivoli Monitoring home directory
– <tems_name>: Tivoli Enterprise Monitoring Server name

Note: <tems_name> is the monitoring server name, not necessarily the Tivoli Enterprise Monitoring Server host name.

The primary TEMS is configured as a Hub(**LOCAL*). All IBM Tivoli Monitoring installations require at least one TEMS configured as a Hub. Additional Remote(**REMOTE*) TEMS can be installed later to introduce a scalable hierarchy into the architecture.

This Hub/Remote interconnection provides a hierarchical design that enables the Remote TEMS to control and collect its individual agent status and propagate the agent status up to the Hub TEMS. This mechanism enables the Hub TEMS to maintain infrastructure-wide visibility of the entire environment. This visibility is passed to the Tivoli Enterprise Portal Server for preformatting, ultimately displaying in the Tivoli Enterprise Portal client.

If security validation is configured, a separate procedure is necessary to manage the OS level user IDs at the Hub TEMS. User access is managed within IBM Tivoli Monitoring 6.1 only via the TEP GUI. A matching user ID must be defined at the HUB TEMS using the standard user management process for that HUB TEMS operating system.

► Tivoli Enterprise Portal Server (TEPS)

The Tivoli Enterprise Portal Server (referred to as the *portal server*) is a repository for all graphical presentation of monitoring data. The portal server database also consists of all user IDs and user access controls for the monitoring workspaces. The TEPS provides the core presentation layer, which allows for retrieval, manipulation, analysis, and preformatting of data. It manages this access through user workspace consoles. The TEPS keeps a persistent connection to the Hub TEMS, and can be considered a logical gateway between the Hub TEMS and the Tivoli Enterprise Portal client. Any disconnection between the two components immediately disables access to the monitoring data used by the Tivoli Enterprise Portal client.

An RDBMS must be installed on the same physical system prior to the TEPS installation. This prerequisite is necessary because the TEPS installation will create the mandatory TEPS database, along with the supporting tables. Additionally, an ODBC (Open Database Connectivity) Data Source Name is configured to connect directly to the Tivoli Data Warehouse RDBMS. This OBDC connection is used whenever a pull of historical data from the Tivoli Data Warehouse is requested.

Note: Even though technically valid, implementing a *remote* RDBMS for the TEPS is not recommended. The TEPS is closely coupled to the RDBMS and the complexity of a remote RDBMS is difficult to maintain.

When installing the TEPS, a proprietary integrated Web server is installed for use with the Tivoli Enterprise Portal client in browser mode. Depending on the network topology and possible security implications, this may play a role in constructing the solution. Instead, an external Web server installed on the same system as the TEPS can be used. For additional details, refer to *IBM Tivoli Monitoring Installation and Setup Guide*, GC32-9407.

In large installations, installing multiple TEPS that connect to one single Hub TEMS is recommended. See "Large installation (4000 agents maximum)" on page 28 for further details.

- Tivoli Enterprise Portal (TEP)

  The TEP client (referred to as the *portal client*) is a Java-based user interface that connects to the Tivoli Enterprise Portal Server to view all monitoring data collections. It is the user interaction component of the presentation layer. The TEP brings all of these views together in a single window so you can see when any component is not working as expected. The client offers two modes of operation: a Java™ desktop client and an HTTP browser.

  Assuming a default installation, the browser-mode TEP client can be found using this URL:

  ```
  http://<hostname>:1920///cnp/kdh/lib/cnp.html
  ```

  Here, <hostname> is the host name of the Tivoli Enterprise Portal Server.

Important: IBM Tivoli Monitoring supports only Internet Explorer on the Windows platform in browser mode.

The following products will have integrated interfaces into TEP:

- OMEGAMON Z
- OMEGAMON Distributed
- IBM Tivoli Monitoring 5.1.2
- IBM Tivoli Monitoring 6.1
- NetView for z/OS (release 5.2)
- IBM Tivoli Enterprise Console
- IBM Tivoli Composite Application Manager for Response Time Tracking
- IBM Tivoli Composite Application Manager for WebSphere
- IBM Tivoli Composite Application Manager for SOA

> **Note:** In 2006, additional products such as IBM Tivoli Service Level Advisor, System Automation, and Tivoli Business System Manager will also be integrated into the Tivoli Enterprise Portal. IBM Tivoli Service Level Advisor integrations will be available with Tivoli Data Warehouse V2.1.1.

- Tivoli Enterprise Management Agent (TEMA)

  The agents (referred to as *managed systems*) are installed on the system or subsystem requiring data collection and monitoring. The agents are responsible for data gathering and distribution of attributes to the monitoring servers, including initiating the heartbeat status.

  These agents test attribute values against a threshold and report these results to the monitoring servers. The TEP displays an alert icon when a threshold is exceeded or a value is matched. The tests are called *situations*.

  What prompts the monitoring server to gather data samples from the agents?

  - Opening or refreshing a workspace that has data views (table or chart views)

    When this happens, the TEPS sends a sampling request to the Hub TEMS. The request is passed to the monitoring agent (if there is a direct connection) or through the Remote TEMS to which the monitoring agent connects. The monitoring agent takes a data sampling and returns the results through the monitoring server and portal server to the portal workspace.

  - The sampling interval for a situation (a test taken at your monitored systems)

    The situation can have an interval as often as once per second or as seldom as once every three months. When the interval expires, the monitoring server requests data samples from the agent and compares the returned values with the condition described in the situation. If the values meet the condition, the icons change on the navigation tree.

Optionally, the agents can be configured to transfer data collections directly to the Warehouse Proxy agent instead of using the Remote TEMS. If firewall restrictions are disabled or minimum, you should configure all the agents to transfer directly to Warehouse Proxy agent. Otherwise, firewall security is a key factor in the location of the Warehouse Proxy agent respective to the firewall zone and agents. Warehousing data through the Remote TEMS is limited and should be used only as a last resort.

Tivoli Enterprise Management Agents are grouped into two categories:

- Operating System (OS) Agents

  Operating System Agents retrieve and collect all monitoring attribute groups related to specific operating system management conditions and associated data.

- Application Agents

  Application Agents are specialized agents coded to retrieve and collect unique monitoring attribute groups related to one specific application. The monitoring groups are designed around an individual software application, and they provide in-depth visibility into the status and conditions of that particular application.

Common management agents packaged with IBM Tivoli Monitoring include:

- Window OS Agent
- Linux OS Agent
- UNIX OS Agent
- UNIX Log Agent
- i5 OS Agent
- Universal Agent

  The Universal Agent is a special agent that leverages a full Application Programming Interface (API) to monitor and collect data for any type of software. Any application that produces data values, the Universal Agent can monitor and retrieve data from it. Essentially, IBM Tivoli Monitoring can now monitor any unique application regardless of whether the base product supports it.

Common optional management agents that are packaged separately include:

- Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint
- DB2® Agent

  - Oracle Agent
  - MS SQL Agent
  - MS Exchange Agent
  - Active Directory Agent
- Warehouse Proxy agent

  The Warehouse Proxy agent is a unique agent that performs only one task: collecting and consolidating all Historical Data Collections from the individual agents to store in the Tivoli Data Warehouse. If using the Tivoli Data Warehouse, one Warehouse Proxy agent is required for each IBM Tivoli Monitoring installation. It uses ODBC (Open Database Connectivity) to write the historical data to a supported relational database.

  Restriction: IBM Tivoli Monitoring currently supports only the Warehouse Proxy agent under the Windows platform. A post-GA release of IBM Tivoli Monitoring will include UNIX operating support.

- Warehouse Summarization and Pruning agent (S&P)

  The Summarization and Pruning agent is a unique agent that performs the aggregation and pruning functions for the historical raw data on the Tivoli Data Warehouse. It has advanced configuration options that enable exceptional customization of the historical data storage.

  One S&P is recommended to manage the historical data in the Tivoli Data Warehouse. Due to the tremendous amounts of data processing necessary, it is recommended the S&P be always installed on the same physical system as the Tivoli Data Warehouse repository.

- Tivoli Data Warehouse (TDW)

  The Tivoli Data Warehouse is the database storage that contains all of the Historical Data Collection. A Warehouse Proxy must be installed, to leverage the TDW function within the environment. In large-scale deployments, a Tivoli Data Warehouse can be shared among monitoring installations.

An IBM Tivoli Monitoring installation can contain these optional components:

- Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint

  Also called IBM Tivoli Monitoring 5.x Endpoint Agent, this integration agent enables the collection and visualization of IBM Tivoli Monitoring 5.x resource models in the Tivoli Enterprise Portal. The visualization is the direct replacement for the Web Health Console. Additionally, the Agent provides roll-up function into the Tivoli Data Warehouse.

- Tivoli Enterprise Console event synchronization

  The TEC event synchronization component sends updates to situation events back to the monitoring server that are forwarded to the event server. Actions performed at the Tivoli Enterprise Console for IBM Tivoli Monitoring situations are reflected in the Tivoli Enterprise Portal Server.

- IBM Tivoli Business Systems Manager (TBSM)

  IBM Tivoli Business Systems Manager provides intelligent management software to help businesses increase operational agility by aligning IT operations to business priorities. Intelligent management software helps optimize IT operations according to the business goals of the organization, rather than focusing on the technology itself.

Note: IBM will provide a special program called TBSM feed from OMEGAMON (or XE Feed) for IBM Tivoli Monitoring and IBM Tivoli Business Systems Manager integration. The XE Feed is planned to be made available as an LA fix to IBM Tivoli Business Systems Manager V3.1 in the first quarter of 2006, then rolled into the IBM Tivoli Business Systems Manager V3.2 release, which is scheduled for September 2006.

2.1.1 Platform support matrix for IBM Tivoli Monitoring

To get most up-to-date information about the platform support matrix for IBM Tivoli Monitoring 6.1, please refer to the following link:

http://www-306.ibm.com/software/sysmgmt/products/support/Tivoli_Supported_Platforms.html

2.1.2 Database support matrix

Table 2-1 shows the database support matrix for IBM Tivoli Monitoring 6.1.

Note: Database names and versions not listed in this table are not supported, including DB2 on mainframes (zLinux, OS/390, z/OS, and so forth).

Table 2-1 Database support matrix

Database name	TEPS[1]	Data Warehouse
DB2 8.1	A	A
DB2 8.2	A	A
MS SQL 2000	A	A

Database name	TEPS[1]	Data Warehouse
Oracle 9.2	D	A
Oracle 10.1	D	A

1. Key: A – Indicates that the platform is supported.
D – Indicates that the platform will not be supported in this release, but may be supported in a later release.

2.2 IBM Tivoli Monitoring deployment scenarios

Deployment scenarios attempt to provide realistic understanding of architecture design. These scenarios should be used mainly for guidance to assist in the planning and deployment strategy used for a production installation, as every deployment strategy is unique and only proper planning can guarantee a successful implementation.

We cover four types of environments:

- "Demo installation (single machine)" on page 25
- "Small/medium installation (400 agents maximum)" on page 26
- "Large installation (4000 agents maximum)" on page 28
- "Huge installation (greater than 4000 agents)" on page 31

Note: Our classification here is based on the number of IBM Tivoli Monitoring 6.1 agents. In practice, sometimes the number of employees is used to define the size of a business; for example, companies with up to 1000 employees are considered as small-to-medium businesses.

Figure 2-2 depicts the interconnections of the various components at their simplest. Other chapters in this book explain the interconnections in further detail. Any limitation with hardware or software is noted in the later chapters.

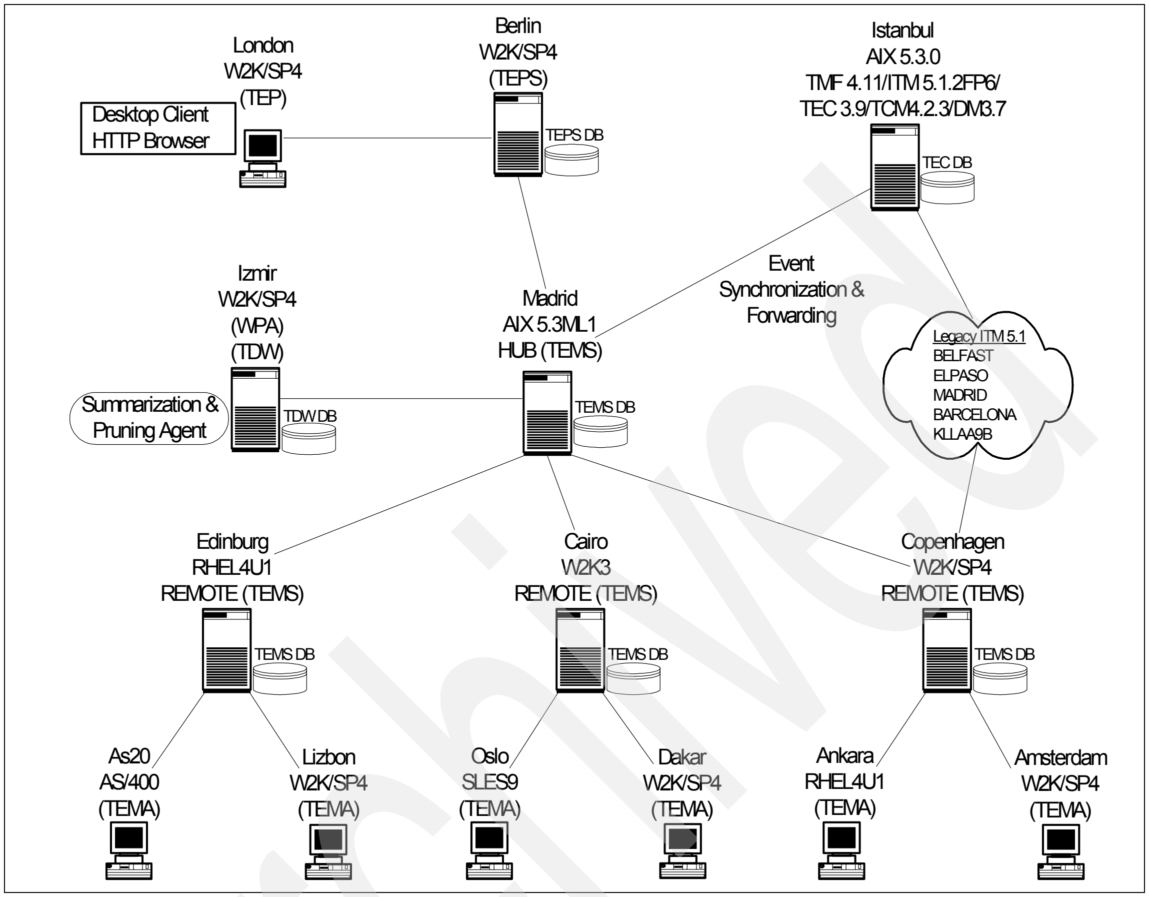


Figure 2-2 IBM Tivoli Monitoring, lab topology

Notes:

- The Hot Standby system is Milan (AIX 5.3.0), which is not depicted in the diagram.
- All of the TEMAs contain at least the OS Agent, and several have additional agents.

To cover various topics throughout this book's development, we implemented an IBM Tivoli Monitoring installation that incorporates all related content. This architecture covers all components that make up an IBM Tivoli Monitoring installation, including the built-in *Hot Standby* Hub Tivoli Enterprise Manager Server. Also, a legacy Tivoli Management Framework V4.1.1 connects to the infrastructure to demonstrate interoperability among IBM Tivoli Monitoring, IBM

Tivoli Monitoring V5.1.2 Fix Pack 6, IBM Distributed Monitoring V3.7, and IBM Tivoli Enterprise Console V3.9. To ensure the accuracy of the implementation and best practices, the environment contains a proportionate selection of heterogeneous hardware configurations with varying degrees of operating system platforms and levels.

Attention: All capacity values, especially for the Tivoli Enterprise Management Agents, are based on approximation. The section headers below provide a recommended maximum number of agents. Also, we include an estimate of the maximum amount of physical systems within the paragraphs that do not calculate out evenly. All these numbers are based on proportionate amounts of agents deployed to every system. Actual production installations may vary greatly in agent disbursement.

2.2.1 Demo installation (single machine)

For demonstration purposes, IBM Tivoli Monitoring can be installed onto a single machine running Windows XP. *This IBM Tivoli Monitoring installation should be used only for demonstration, and is not a supported implementation.* Using the Windows install shield, IBM Tivoli Monitoring can be installed using the single CD. The minimum required software is:

- Tivoli Enterprise Monitoring Server (TEMS)
- Tivoli Enterprise Portal Server (TEPS)
- Tivoli Enterprise Portal Client (TEP)
- Windows OS Agent

Optionally, the Tivoli Warehouse Proxy, Tivoli Data Warehouse, Summarization and Pruning agent, and a DB2 installation can be installed on the same system to illustrate the Historical Data Collection features of IBM Tivoli Monitoring.

Note: IBM Tivoli Monitoring demo (on a single Windows machine) installation will be covered in another book, named *Deployment Guide Series: IBM Tivoli Monitoring V 6.1*, SG24-7188. This redbook will also cover some proof of concept scenarios. Planned availability is December 2005.

2.2.2 Small/medium installation (400 agents maximum)

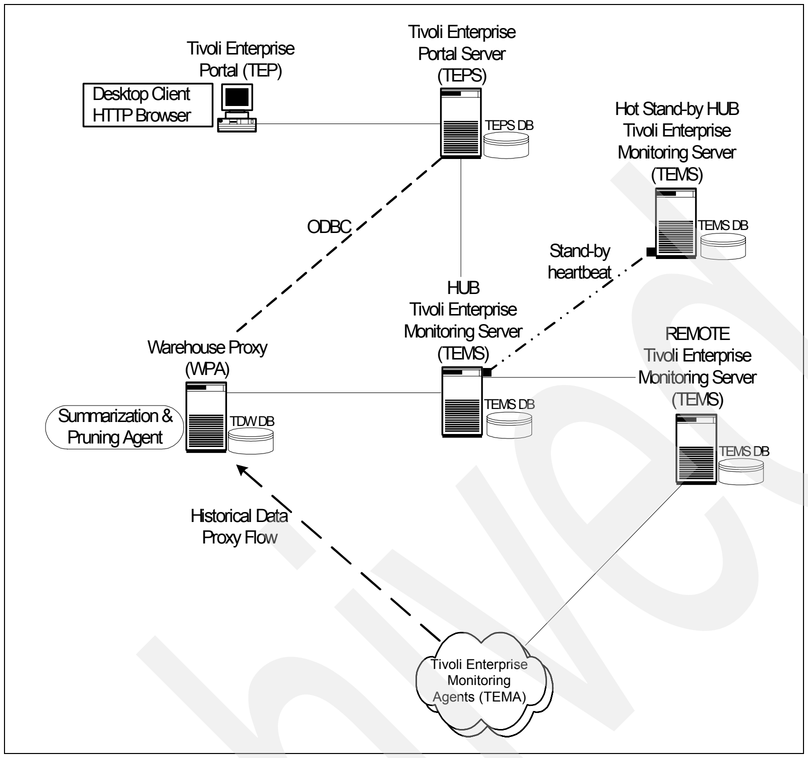
The small/medium installation is the fundamental design utilizing only the minimum required components. This scenario is perfect for prototyping IBM Tivoli Monitoring or using it within a production installation consisting of 400 agents. In fact, IBM Tivoli Monitoring 6.1 by design excels in superiority for the small/medium installation. The out-of-box monitoring collections, GUI presentation layer, Historical Data Collection, and robustness provide a full monitoring solution with a modest total cost of ownership (TCO).

It is implemented with the minimum hardware requirements necessary for a production IBM Tivoli Monitoring installation.

The installation consists of the following components:

- Tivoli Enterprise Monitoring Server
- Tivoli Enterprise Portal Server
- Tivoli Enterprise Portal
- Tivoli Warehouse Proxy agent
- Tivoli Data Warehouse
- Summarization and Pruning agent

Figure 2-3 on page 27 depicts the small/medium topology. The diagram provides an overview of each IBM Tivoli Monitoring connected component. For a comprehensive architecture, the optional Hot Standby node is depicted in this diagram.



Tivoli Enterprise Portal (TEP)
Desktop Client HTTP Browser
Tivoli Enterprise Portal Server (TEPS)
TEPS DB
Hot Stand-by HUB Tivoli Enterprise Monitoring Server (TEMS)
TEMS DB
ODBC
Stand-by heartbeat
HUB Tivoli Enterprise Monitoring Server (TEMS)
Warehouse Proxy (WPA)
REMOTE Tivoli Enterprise Monitoring Server (TEMS)
TEMS DB
Summarization & Pruning Agent
TDW DB
TEMS DB
Historical Data Proxy Flow
Tivoli Enterprise Monitoring Agents (TEMA)

Figure 2-3 IBM Tivoli Monitoring, small/medium topology design

We recommend installing at least three TEMS (including the Hot Standby node) in this scenario, even though the small/medium installation allows the use of only one TEMS. Implementing a Hub/Remote architecture in the early stages allows for growth and scalability. Furthermore, this design builds around IBM Tivoli Monitoring built-in failover capabilities. The small/medium installation supports approximately 250 managed systems. This estimate assumes that the managed systems will have two agents each. The actual distribution of agents will not necessarily be proportionate in a real installation, but this calculation provides the recommended total amount for one IBM Tivoli Monitoring installation. All of the agents will connect to the Remote TEMS using the Hub TEMS as a failover monitoring server.

Optionally, you can install the Hot Standby node, This is recommended but not required for the small/medium installation, especially if cost restrictions exist for hardware deployment. The Hot Standby should always be considered because it offers failure protection with minimum increase in total cost of ownership.

Attention: A small/medium installation cannot use a Remote TEMS as a Hot Standby node. Hot Standby nodes always must be configured as *LOCAL.

Although it can handle agent tasks directly, we do not recommend using the Hub TEMS for this purpose. Rather, it should focus on data collecting and processing tasks between the TEPS and itself. If the environment expands, additional Remote TEMS should be installed to process the additional agent requirement. Additional agent deployments increase processing requirements for the Hub TEMS, which can degrade if the Hub is allowed to handle agent tasks directly.

For an average Tivoli Data Warehouse installation in a small/medium installation, having the Warehouse Proxy agent and the Tivoli Data Warehouse repository on the same system should be sufficient. This installation provides Historical Data Collection without the additional hardware. It is still a wise decision to monitor the Tivoli Data Warehouse after installation to ensure processing rate is on target.

2.2.3 Large installation (4000 agents maximum)

Building on the fundamentals of the small/medium installation, the large installation focuses on scalability. This Tivoli Monitoring environment consists of 4000 agents within a single Tivoli Monitoring installation. It requires the recommended hardware specification or higher to properly scale the infrastructure.

The installation consists of the following components:

- Tivoli Enterprise Monitoring Server
- Tivoli Enterprise Portal Server
- Tivoli Enterprise Portal
- Tivoli Warehouse Proxy agent
- Tivoli Data Warehouse
- Summarization and Pruning agent
- Tivoli Enterprise Console

Figure 2-4 on page 29 depicts the comprehensive architecture for all interconnected components. It points out the recommended strategy for the Tivoli historical date collection. We highly advise structuring the historical collection flow as outlined in the diagram.

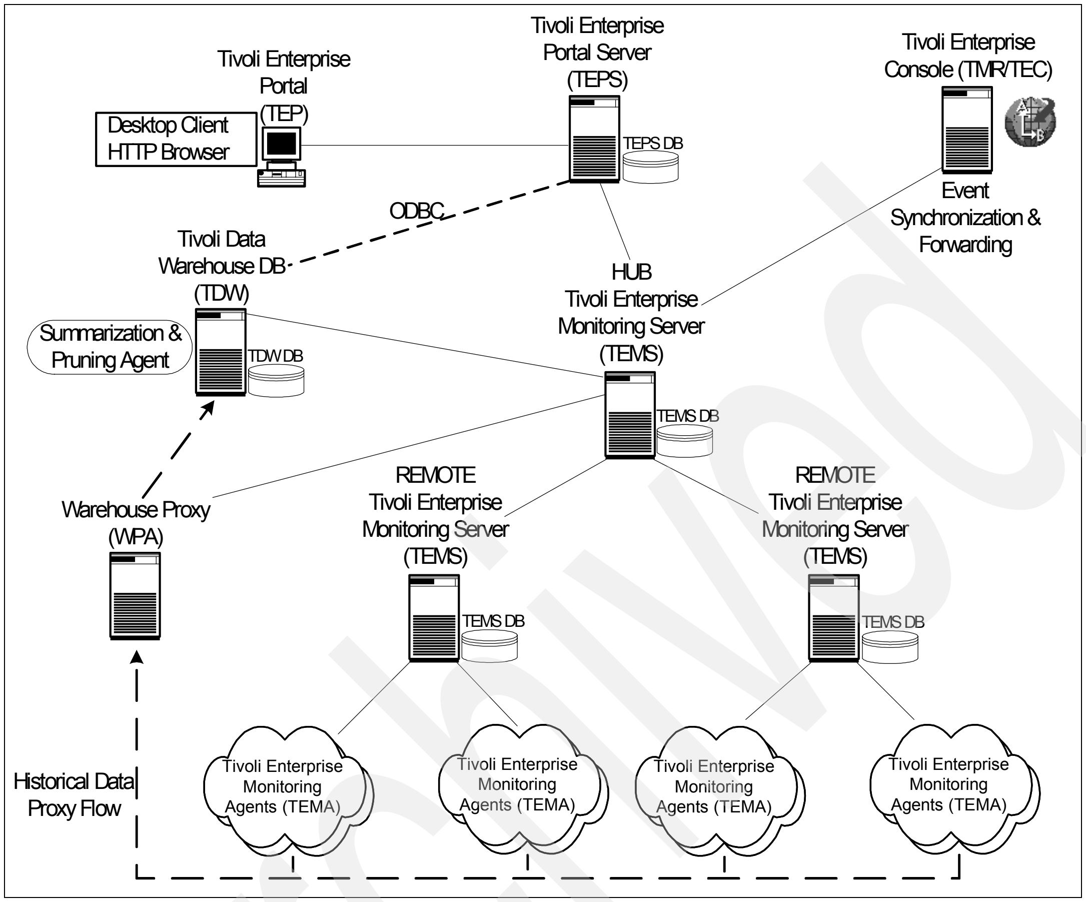

Tivoli Enterprise Portal (TEP)
Desktop Client HTTP Browser
Tivoli Enterprise Portal Server (TEPS)
TEPS DB
Tivoli Enterprise Console (TMR/TEC)
Event Synchronization & Forwarding
ODBC
Tivoli Data Warehouse DB (TDW)
TDW DB
Summarization & Pruning Agent
HUB Tivoli Enterprise Monitoring Server (TEMS)
TEMS DB
Warehouse Proxy (WPA)
REMOTE Tivoli Enterprise Monitoring Server (TEMS)
TEMS DB
REMOTE Tivoli Enterprise Monitoring Server (TEMS)
TEMS DB
Historical Data Proxy Flow
Tivoli Enterprise Monitoring Agents (TEMA)
Tivoli Enterprise Monitoring Agents (TEMA)
Tivoli Enterprise Monitoring Agents (TEMA)
Tivoli Enterprise Monitoring Agents (TEMA)

Figure 2-4 IBM Tivoli Monitoring large topology design

Important: For simplicity, the Hot Standby node is not shown in the topology diagram. In a large installation, implementing the Hot Standby node is strongly recommended.

Performing an accurate plan and assessment stage is imperative for the large installation. Mapping all component topology with the recommended hardware specifications is critical in order to achieve a highly distributed environment with realistic goals. We recommend having a thorough understanding of the monitoring environment before preceding to implement any architectural design. It is important to account for all variables within the topology. Substantial consideration should be given to the infrastructure hardware requirements and the underlying network topology. Network bandwidth, latency, and firewall restriction all require assessment.

IBM Tivoli Monitoring is ideal for small/medium installations. After installation, it begins leveraging the best practice functionality immediately. Default situations start running, and if Historical Data Collection is turned on, the default attribute groups begin analysis and warehousing. These default services can impede the large installation performance throughput, especially if unnecessary attributed group collections are enabled. We highly suggest changing the *Run at Startup* property on all situations to `NO` immediately after the TEMS, TEPS, and TEP are deployed. This practice ensures the freedom to execute the business plan strategy (defining managed system list, customized situation, event mapping, date warehousing intervals, and so forth) that are generated from the assessment and planning phrase. It is vital to the health of the large installation that only the desired situations and attribute groups are enabled.

A large monitoring installation supports approximately 1,500 managed systems in an environment. For the large installation, the estimate is three agents per managed system. In this installation, a disproportionate distribution of agents is highly anticipated, and this scenario should complement your own environment analysis phrase. The recommended distribution is 400 agents across 10 Remote TEMSs. Keeping 400 agents as the high point per monitoring server allows for capacity expansion without exhausting the resources of the infrastructure. For further details about scaling a large installation, refer to “Scalability” on page 48.

Tip: Because IBM Tivoli Monitoring supports primary and secondary communication paths, we suggest installing several backup Remote TEMSs that exist solely for TEMA failover capabilities. If a Remote TEMS fails, we do not advise doubling the maximum load of production Remote TEMSs. Best practices should direct these orphan Tivoli Enterprise Management Agents to idle Remote TEMS.

The Tivoli Data Warehouse data requirement will be substantial. We advise separating the Tivoli Warehouse Proxy agent and the Tivoli Data Warehouse repository between two systems. The Summarization and Pruning agent should be installed on the Tivoli Data Warehouse system. We always recommend keeping these two components together.

The large installation introduces the IBM Tivoli Enterprise Console as part of the topology. IBM Tivoli Monitoring has built-in capabilities for event processing that work extremely well in the small/medium installation. However, the large installation can contain a reasonable increase in volume of event flow, and the Tivoli Enterprise Console is better adapted for large event flow management and correlation. The Tivoli Enterprise Console can be considered an event consolidation Manager of Managers.

The TCO is still nominal compared to IBM Tivoli Monitoring functionality, despite the large hardware requirements needed to scale this installation properly. The entire large installation can be managed from a single GUI presentation layer down to installing and upgrading agents.

2.2.4 Huge installation (greater than 4000 agents)

The huge installation scenario provides a guideline for any IBM Tivoli Monitoring installation that exceeds 4000 agents, or approximately 1,500 managed systems. The scope of the huge installation is similar to the large installation, except for additional configuration guidance.

The installation consists of the following components:

- Tivoli Enterprise Monitoring Server
- Tivoli Enterprise Portal Server
- Tivoli Enterprise Portal
- Tivoli Warehouse Proxy agent
- Tivoli Data Warehouse
- Summarization and Pruning agent
- Tivoli Enterprise Console

Figure 2-5 on page 32 depicts the interconnections between two autonomous IBM Tivoli Monitoring installations. It demonstrates the high-level component interaction between two installations that handle 4,000 agents each, totaling 8,000 agents entirely.

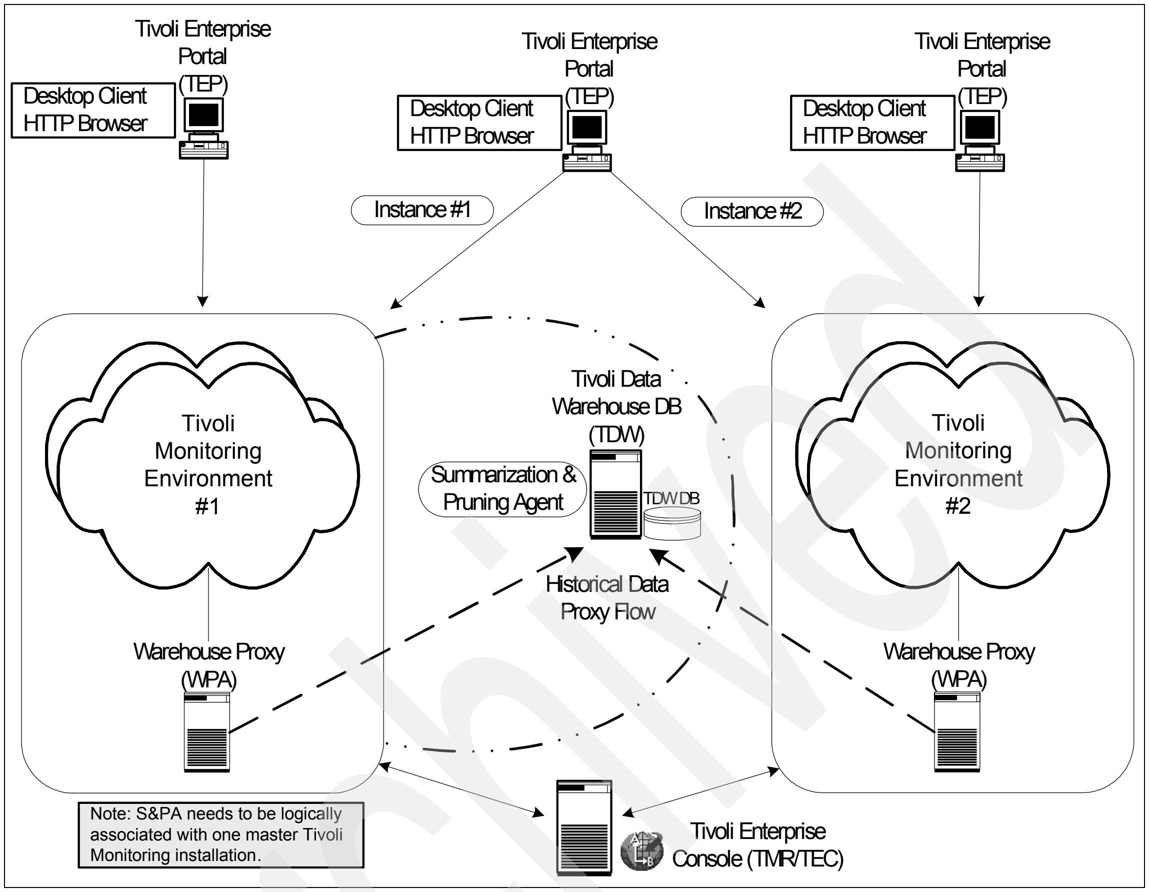

Tivoli Enterprise Portal (TEP)
Desktop Client HTTP Browser
Tivoli Enterprise Portal (TEP)
Desktop Client HTTP Browser
Tivoli Enterprise Portal (TEP)
Desktop Client HTTP Browser
Instance #1
Instance #2
Tivoli Monitoring Environment #1
Tivoli Data Warehouse DB (TDW)
Summarization & Pruning Agent
TDW DB
Tivoli Monitoring Environment #2
Historical Data Proxy Flow
Warehouse Proxy (WPA)
Warehouse Proxy (WPA)
Note: S&PA needs to be logically associated with one master Tivoli Monitoring installation.
Tivoli Enterprise Console (TMR/TEC)

Figure 2-5 IBM Tivoli Monitoring huge installation topology

The recommended deployment strategy is the same as for the large installation, except for the Tivoli Data Warehouse, and Summarization and Pruning agent. A huge installation can warehouse Historical Data Collections to one single database server repository from two distinct IBM Tivoli Monitoring installations.

Important: As noted in "Large installation (4000 agents maximum)" on page 28, make sure that only the required attributed groups are enabled for Tivoli Data Warehousing. Enormous amounts of data can be collected between two large IBM Tivoli Monitoring installations. Best practice design is critical to ensure a stable, scalable environment.

The two installations are still built separately from each other. The only deviation is that one IBM Tivoli Monitoring installation requires a logical association as the *master* control for the Summarization and Pruning agent.

> **Note:** There can be only one Summarization and Pruning agent for a single Tivoli Data Warehouse. Because the Summarization and Pruning agent requires connections to a TEMS, one of the monitoring installations must be logically designated as the master. This is not a programmatic assignment, but a logical identification for configuration and management of the S&P.

A flexible feature that is needed in the huge installation is the ability to configure multiple TEP instances in a single TEP desktop client. If a single TEP desktop client has to connect to a separate autonomous IBM Tivoli Monitoring installation, *instances* are created to associate the unique TEPS connection information.

Defining TEP instances via Tivoli Manage Service GUI

Use the following steps in the Manage Tivoli Enterprise Monitoring Services GUI to define TEP instances for additional Hub TEMS.

1. Start the Manage Tivoli Enterprise Monitoring Services GUI.

Windows	Click **Start** → **Programs** → **IBM Tivoli Monitoring** → **Manage Tivoli Enterprise Monitoring Services**.
UNIX/Linux	Type `itmcmd manage`

2. Right-click the Tivoli Enterprise Portal and click **Create Instance** as shown in Figure 2-6 on page 34.

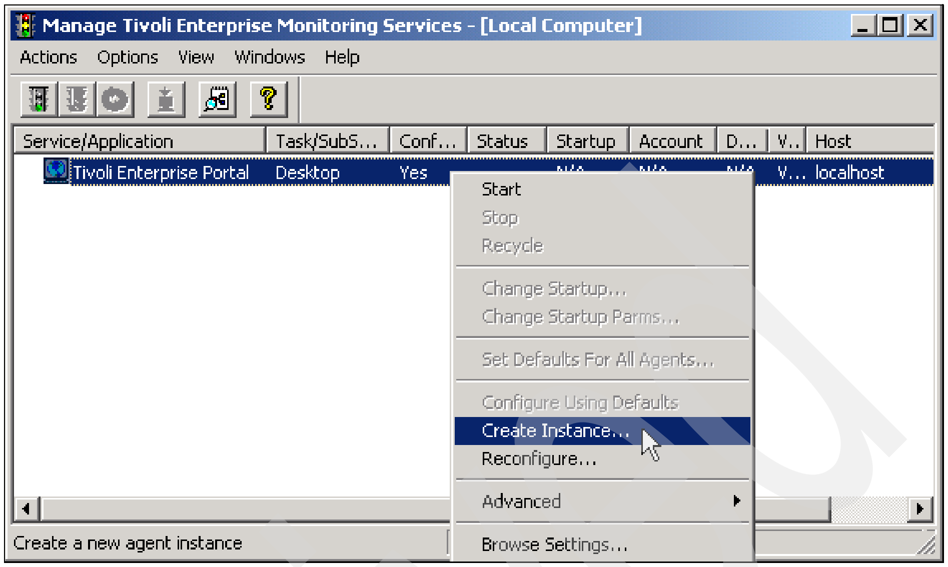

Manage Tivoli Enterprise Monitoring Services - [Local Computer]
Actions Options View Windows Help
Service/Application Task/SubS... Conf... Status Startup Account D... V.. Host
Tivoli Enterprise Portal Desktop Yes N/A N/A N/A V... localhost
Start
Stop
Recycle
Change Startup...
Change Startup Parms...
Set Defaults For All Agents...
Configure Using Defaults
Create Instance...
Reconfigure...
Advanced
Browse Settings...
Create a new agent instance

Figure 2-6 Right-click Tivoli Enterprise Portal for Create Instance option

3. Type the instance name and click **OK** (Figure 2-7).

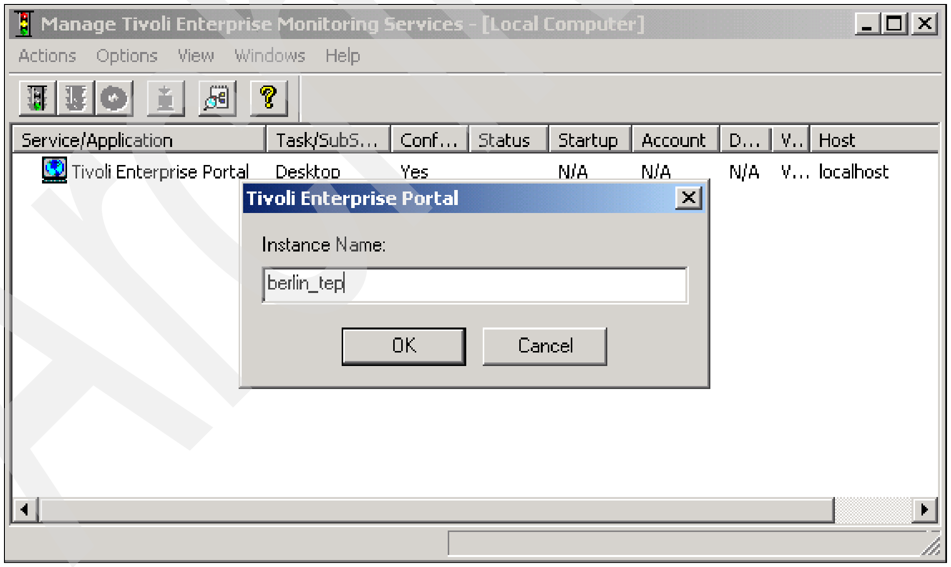
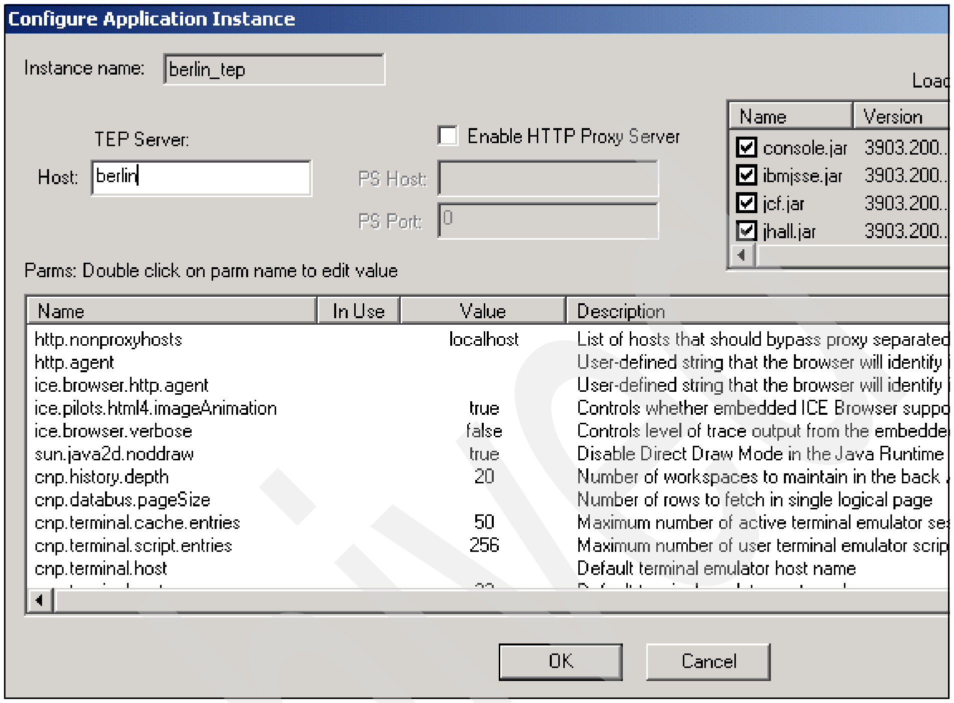

Manage Tivoli Enterprise Monitoring Services - [Local Computer]
Actions Options View Windows Help
Service/Application Task/SubS... Conf... Status Startup Account D... V.. Host
Tivoli Enterprise Portal Desktop Yes N/A N/A N/A V... localhost
Tivoli Enterprise Portal
Instance Name:
berlin_tep
OK
Cancel

Figure 2-7 Entering the Instance Name into the dialog box

4. Type the Tivoli Enterprise Portal host name and click **OK** (Figure 2-8**)**.



Instance name: berlin_tep

TEP Server:

Host: berlin

PS Host:

PS Port: 0

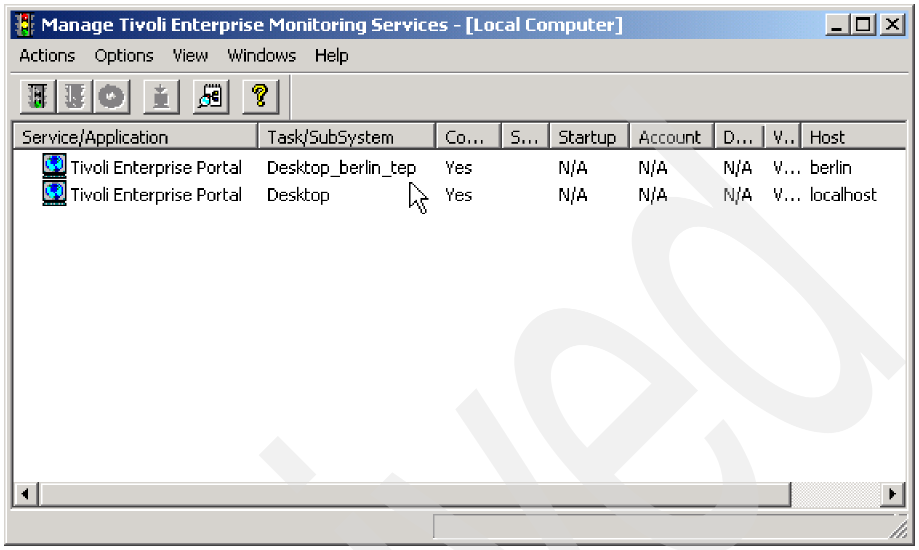
Name	Version
✓ console.jar	3903.200..
✓ ibmjsse.jar	3903.200..
✓ jcf.jar	3903.200..
✓ jhall.jar	3903.200..

Parms: Double click on parm name to edit value

Name	In Use	Value	Description
http.nonproxyhosts		localhost	List of hosts that should bypass proxy separated
http.agent			User-defined string that the browser will identify
ice.browser.http.agent			User-defined string that the browser will identify
ice.pilots.html4.imageAnimation		true	Controls whether embedded ICE Browser suppo
ice.browser.verbose		false	Controls level of trace output from the embedde
sun.java2d.noddraw		true	Disable Direct Draw Mode in the Java Runtime
cnp.history.depth		20	Number of workspaces to maintain in the back
cnp.databus.pageSize			Number of rows to fetch in single logical page
cnp.terminal.cache.entries		50	Maximum number of active terminal emulator se
cnp.terminal.script.entries		256	Maximum number of user terminal emulator scrip
cnp.terminal.host			Default terminal emulator host name

Figure 2-8 Entering Tivoli Enterprise Portal host name into TEP Server field

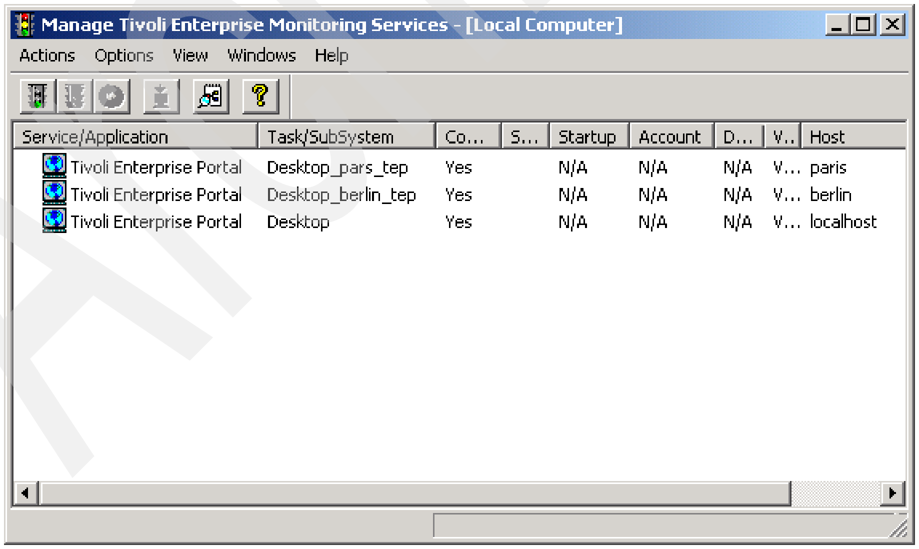
5. The new Tivoli Enterprise Portal is now displayed in the Manage Tivoli Enterprise Monitoring GUI (Figure 2-9).


Manage Tivoli Enterprise Monitoring Services - [Local Computer]
Actions Options View Windows Help

Service/Application	Task/SubSystem	Co...	S...	Startup	Account	D...	V...	Host
Tivoli Enterprise Portal	Desktop_berlin_tep	Yes		N/A	N/A	N/A	V...	berlin
Tivoli Enterprise Portal	Desktop	Yes		N/A	N/A	N/A	V...	localhost

Figure 2-9 The newly defined Tivoli Enterprise Portal instance

Subsequent Tivoli Enterprise Portal instances are defined repeating steps 1 - 4 (Figure 2-10).


Manage Tivoli Enterprise Monitoring Services - [Local Computer]
Actions Options View Windows Help

Service/Application	Task/SubSystem	Co...	S...	Startup	Account	D...	V...	Host
Tivoli Enterprise Portal	Desktop_pars_tep	Yes		N/A	N/A	N/A	V...	paris
Tivoli Enterprise Portal	Desktop_berlin_tep	Yes		N/A	N/A	N/A	V...	berlin
Tivoli Enterprise Portal	Desktop	Yes		N/A	N/A	N/A	V...	localhost

Figure 2-10 Example of additional Tivoli Enterprise Portal instances

2.2.5 Advanced large installation with firewall scenarios

In most IBM Tivoli Monitoring implementations, firewalls play an important role throughout the architecture. For a successful implementation, it is important to understand the component communication flow. The configuration to support IBM Tivoli Monitoring within firewalls has two major parts:

- The TEMS, TEPS, and TEMA protocol communication
- The TEP and TEPS protocol communication

Tip: Refer to the *IBM Tivoli Monitoring Installation and Setup Guide*, GC32-9407, for expert advice about firewall scenarios. This book has several excellent examples using firewalls involving the TEP and TEPS.

Communications protocol selection

If installing IBM Tivoli Monitoring components across firewalls, the recommendation is to configure the *IP.PIPE* (TCP communication) protocol. The *IP* (UDP communication) protocol is insufficient for firewall configurations. The connectionless UDP protocol requires opening up multiple ports across firewalls to allow multiple connections from each individual IBM Tivoli Monitoring component. For example, a TEMA communicating to the TEMS using IP (UDP communication) protocol requires multiple ports to operate properly. Also, using the IP.PIPE (TCP communication) enables the *Ephemeral pipe* operation automatically if certain conditions match.

Note: When IP.PIPE is specified as your communications protocol, you may still see other ports being used in communication traces and logs, but these ports are virtual and multiplexed over the default IP.PIPE port.

The IP.PIPE protocol has some notable limitations:

- Only 16 IBM Tivoli Monitoring processes on a single system can share the base listening port (default port 1918) on a single network interface card when using the protocol. Any processes above 16 will fall back to using the IP protocol (only if configured). This mainly is a restriction when running large numbers of Tivoli Enterprise Management Agents on one physical system. It is not a limitation for the total amount of TEMA connecting to one TEMS. This may occur only when a system is required to run more than 16 Universal Agents or has more than 16 Database Agent instances. If firewall restrictions force the use of the IP.PIPE protocol, the only workaround is to move excess Tivoli Enterprise Management Agents above 16 to another system.
- The TEMS may run out of sockets (listen threads). The TEMS log shows evidence of this:

```
message KDSMA010 – Communication did not succeed.
```

If this occurs, you should increase the number of sockets by changing the setting of KDS_NCSLISTEN. The maximum value that can be set is 256.

Table 2-2 depicts the *default* listening ports for the IBM Tivoli Monitoring components. Use this table as a quick reference to understand the standard ports for an installation. Although modifying these default values is supported, it is not recommended.

Table 2-2 Default port usage for IBM Tivoli Monitoring

IBM Tivoli Monitoring Component	Listening Port
Tivoli Enterprise Monitoring Server (IP.PIPE)	1918/tcp
Tivoli Enterprise Monitoring Server (IP.SPIPE)	3660/tcp
Tivoli Enterprise Monitoring Server (IP)	1918/udp
Tivoli Enterprise Portal Server	1920/tcp 15001/tcp
Tivoli Enterprise Console	5529/tcp
Tivoli Warehouse Proxy agent	6014/tcp[1]

1. Refer to Example 2-1 on page 39.

Tip: Do not deviate from the default listening ports without a valid reason, even though this is supported. Listening port modification was not tested by IBM Tivoli Software Group.

Using IP.PIPE enables a few well-known ports to be open through the firewall. You can use Example 2-1 on page 39 to calculate which port to open. If the firewall is not using NAT (Network Address Translation), the computation should be sufficient to have the components connect through the firewall.

Every system that has IBM Tivoli Monitoring installed will automatically reserve the well-known port (default 1918) for the Tivoli Enterprise Monitoring Server communication. No matter what order components start up on a system that has several IBM Tivoli Monitoring components installed, the default well-known port is only used by the TEMS.

Note: 1918 is the default *well-known* port. Any well-known port can be configured, as long as the entire environment matches this port number.

For all components other than the TEMS, the calculation in Example 2-1 is used internally by IBM Tivoli Monitoring to reserve the listening ports.

Example 2-1 IBM Tivoli Monitoring algorithm to calculate listening port

```
"reserved port" = well-known port + (N*4096)
where:
N= startup sequence
```

For example, the IBM Tivoli Monitoring component startup on the system Izmir follows this sequence:

1. The Universal Agent starts first: port 6014 (1918 + 1*4096)
2. The Remote TEMS starts second: port 1918 (always reserved for TEMS)
3. The Windows OS Agent starts third: port 10110 (1918 + 2*4096)
4. The Warehousing Proxy starts fourth: port 14206 (1918 + 3*4096)

Not all communication is through the firewall

Using the calculation from Example 2-1, it is now possible to control the port usage on individual systems. Additionally, using two parameters in the KDC_FAMILIES environment variable enables even finer control than the startup sequence method. Ideally, all components that need access through the firewall should use the lower-number ports, and components that do not cross the firewall use higher-number ports.

This is accomplished by specifying the SKIP and COUNT parameters on the KDC_FAMILIES environment variable for the individual IBM Tivoli Monitoring component. (See Example 2-2.)

For example:

```
KDC_FAMILIES=IP.PIPE COUNT:1 PORT:1918 IP use:n SNA use:n IP.SPIPE use:n
```

- The `COUNT` parameter (coded as COUNT:N where N is an integer that indicates which port to reserve) for the components that need access across a firewall. If the process is unable to bind to the highest port respective to N, it immediately fails to start up.
- The SKIP parameter (coded as SKIP:N where N is an integer that indicates which port to reserve +1) for the components that do not need access across a firewall. If the process is unable to bind to the port respective to N, it will keep trying using the algorithm until all available ports are exhausted.

Example 2-2 Example for KDC_FAMILIES=IP.PIPE COUNT

```
The system Izmir has installed:
-Tivoli Enterprise Monitoring Server
-Windows OS Agent
-Warehousing Proxy agent
```

```
The well-known port is the default port 1918.
The Tivoli Enterprise Monitoring Server always uses port 1918.
The Windows OS agent does not require firewall access and should be coded with
KDC_FAMILIES=IP.PIPE SKIP:2 (port 10110).
If the Windows OS agent fails to open port 10110, it will try SKIP:3 attempting
to bind now to port 10370. A failure will result in trying SKIP:4 continuing to
exhaust all possibilities with any subsequent failures.
The Warehouse Proxy does require firewall access and should coded with
KDC_FAMILIES=IP.PIPE COUNT:1 (port 6014).
If the Warehouse Proxy fails to open port 6014, start up fails.
```

Multiple network interface cards

Whenever an IBM Tivoli Monitoring component starts up, by default it discovers all available network interfaces on the system and actively uses them. This may not always produce the desired results.

Consider, for example, a TEMS with two networking interface cards (NIC): one interface connected to the main production network and a second interface connected to a limited network that is used only for server backup.

When a TEMA on another system starts up and makes the first connection to the TEMS using the Global Location Broker, it connects to the TEMS first interface. Also, assume that the TEMA does not have an interface connected to the limited backup network segment. The TEMS sends a reply to the TEMA that contains the network address on which the TEMS wants the TEMA to connect. This network address may be the NIC that is connected to the backup network. This results in the TEMA not being able to connect successfully even though the initial handshake succeeded.

To avoid this problem, you can specify an environment parameter on all of the IBM Tivoli Monitoring components to force it to use a specific network interface rather then using any available.

This can be accomplished by passing either of these keywords:

- KDCB0_HOSTNAME: You can specify either the host name, corresponding to the NIC to be used, or its IP address in dotted decimal format. If specified, it will take priority over the KDEB_INTERFACELIST parameter. KDCB0_HOSTNAME should be used only in an environment without NAT (Network Address Translation), as it will also inactivate the use of the Ephemeral Pipe.
- KDEB_INTERFACELIST: The NIC must be specified as dotted decimal IP addresses. This keyword is recommended when IBM Tivoli Monitoring is installed in a environment with NAT.

Regardless, this technique is still a good practice to ensure that the Tivoli Enterprise Management Agents connect to the proper TEMS interface.

Installations with firewalls

The best practice with Tivoli Enterprise Management Agents on the less secure zone of the firewall is to deploy a Remote TEMS on the same firewall side. This enables all TEMAs to connect to the Remote TEMS and have only the Remote TEMS connect through the firewall.

This minimizes the number of systems that need firewall access and keeps port restrictions in place. Refer to Figure 2-11 on page 43 and Figure 2-12 on page 44 for a visual diagram.

Special cases

- Firewall with NAT — Ephemeral Pipe

  Today, many firewall implementations include Network Address Translation, which further protects the systems behind the firewall by making them "invisible" using a different set of IP addresses. If the configuration includes a firewall with NAT, the easiest way to configure TEMA, TEPS, or TEMS to connect to another TEMS would be the Ephemeral Pipe. When an Ephemeral Pipe is active, it acts as a *virtual tunnel* that funnels all connections between two components through one single port. The Ephemeral Pipe is not explicitly started when using the standard installation scripts or tools, but will be activated by default under following conditions:

  – KDC_PARTITION definition file is not present; if KDC_PARTITION is used, it inactivates the Ephemeral Pipe.

  – KDCB0_HOSTNAME parameter should not be specified; instead use the KDEB_INTERFACELIST variable.

  – The initial communication must come from the agents, not by the TEMS. Older configurations may still have a KDSSTART LBDAEMON command for the Location Broker at the TEMS. This command should be removed to active the Ephemeral Pipe.

  If these conditions are met, TEMA-to-TEMS communication automatically tries to create an Ephemeral Pipe connection and no further configuration actions are required. The main advantage of using Ephemeral Pipe is that no special configuration is required, so you do not have to *manually* update the configuration parameters at possibly hundreds of TEMAs that run outside of the firewall.

  The Ephemeral Pipe can be explicitly configured by setting this parameter:

  ```
  KDC_FAMILIES=IP.PIPE PORT 1928 EPHEMERAL:Y
  ```

This forces the client to use outbound ephemeral connections. This kind of configuration should be used if you encounter `duplicate pipe setup failure` messages in the TEMS log – which occurs if you run multiple agents on the same system as the TEMS and all connect to that particular TEMS using the same pipe. In this case, `EPHEMERAL:Y` forces the agents to use the Ephemeral Pipe.

Although Ephemeral Pipe is the first choice for firewall environments with NAT, it may not communicate successfully across firewalls in all environments. If communication failures occur between the TEMA and the TEMS, a more detailed communications trace will be required. Set the KDC_DEBUG=Y variable to generate the required level of detail trace.

If the output of the KDC_DEBUG=Y trace contains IP addresses with 0.0.0.0, this indicates correct use of Ephemeral Pipe. However, if communications is still failing, you will have to use the alternative technique that requires partition definitions. This can happen if the connections between TEMA and TEMS have to cross multiple firewalls or if NAT has been set up without using generic patterns.

- Firewall with NAT – Partitioning

  If Ephemeral Pipe fails to establish a connection between the agents and the Hub TEMS, the only alternative with this IBM Tivoli Monitoring release is to use partition files. This is fully documented in the *IBM Tivoli Monitoring Installation and Setup Guide*, GC32-9407.

Large installation with firewall architectures

Keep in mind that security guidelines in a specific environment may be inflexible when dealing with the location of some of the IBM Tivoli Monitoring components. Accurate comprehension of the communication flows and ports enable any installation to be customized to meet the underlying security policies.

The following recommended architectures provide visual guidance in understanding the communication flow among the IBM Tivoli Monitoring components. Mastery of IBM Tivoli Monitoring communication protocol provides the architect control over the entire network topology. We now describe two common designs.

Warehouse Proxy agent in less secure zone

This scenario is based on less-restrictive firewall rules concerning the traffic flow for the Historical Data Collection. Here the Warehouse Proxy agent is located in the less secure zone.

Figure 2-11 on page 43 depicts one recommended architecture for a IBM Tivoli Monitoring installation with firewall restrictions enabled and with the Warehouse Proxy agent located in the less secure zone.

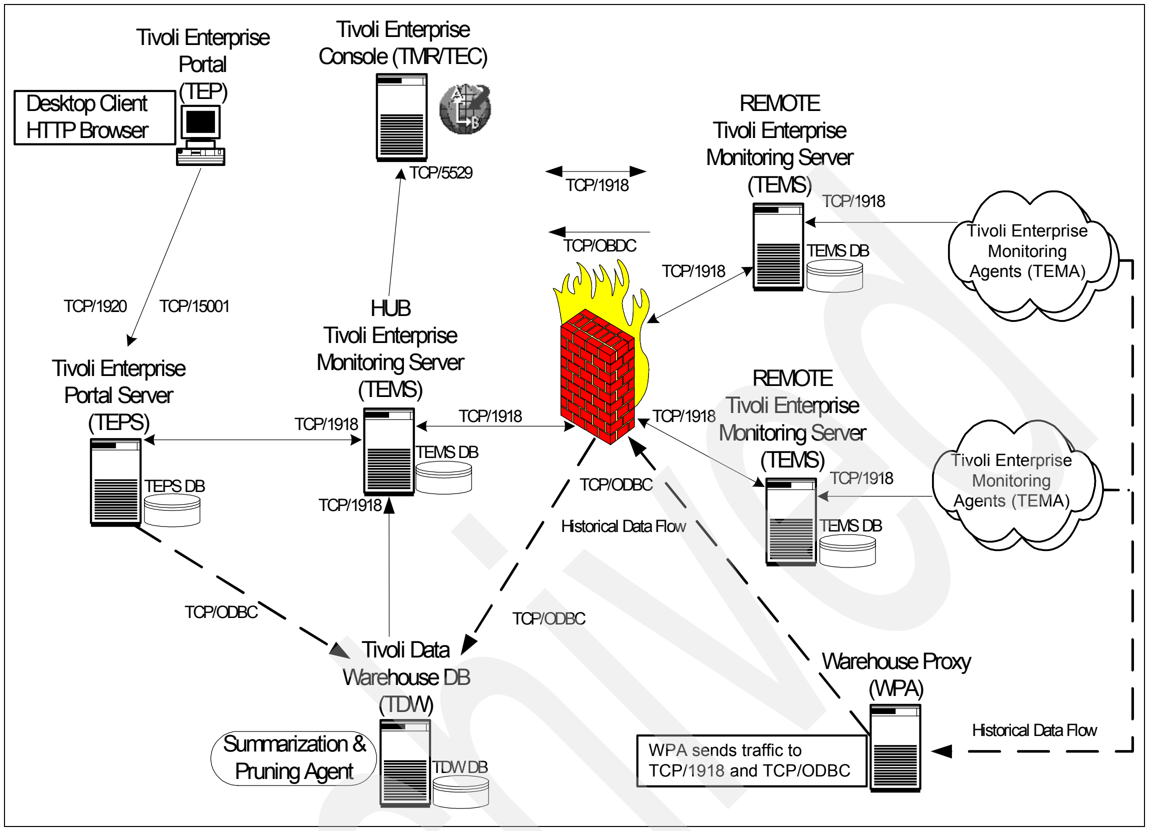

Tivoli Enterprise Portal (TEP)
Desktop Client HTTP Browser
Tivoli Enterprise Console (TMR/TEC)
TCP/5529
TCP/1918
TCP/OBDC
REMOTE Tivoli Enterprise Monitoring Server (TEMS)
TCP/1918
TEMS DB
Tivoli Enterprise Monitoring Agents (TEMA)
TCP/1918
TCP/1920
TCP/15001
HUB Tivoli Enterprise Monitoring Server (TEMS)
Tivoli Enterprise Portal Server (TEPS)
TCP/1918
TCP/1918
TEMS DB
TEPS DB
TCP/1918
TCP/1918
REMOTE Tivoli Enterprise Monitoring Server (TEMS)
TCP/1918
Tivoli Enterprise Monitoring Agents (TEMA)
TEMS DB
TCP/ODBC
Historical Data Flow
TCP/ODBC
TCP/ODBC
Tivoli Data Warehouse DB (TDW)
Warehouse Proxy (WPA)
Summarization & Pruning Agent
TDW DB
WPA sends traffic to TCP/1918 and TCP/ODBC
Historical Data Flow

Figure 2-11 Advanced installation on less secure side

This scenario keeps the TEMA warehousing traffic on the same side of the firewall, and the actual database repository on the more secure side. It is unnecessary to keep track of the Warehouse Proxy agent listening port for firewall rules. A specific port must be opened on the firewall to enable the Warehouse Proxy agent to perform an ODBC connection to the Tivoli Data Warehouse on the more secure side. The port for the ODBC connection is unique to each RDBMS. Consult the database product manuals or your local database administrator.

Warehouse Proxy agent in more secure zone

In this second scenario, the firewall restrictions are expanded to prevent any warehousing traffic on the less secure side of the firewall.

Figure 2-12 on page 44 depicts the recommended architecture for an IBM Tivoli Monitoring installation with firewall restrictions increased, and the Warehouse Proxy agent located in the more secure zone.

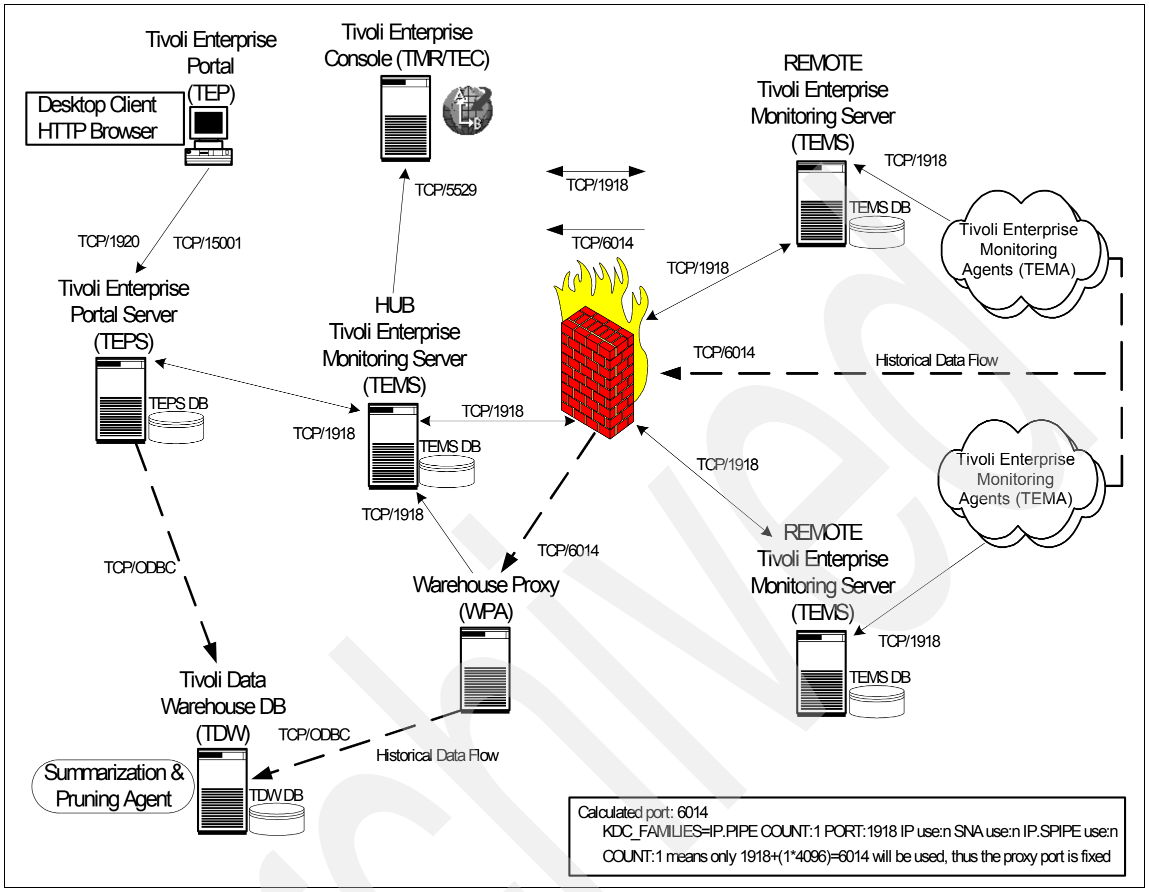

Tivoli Enterprise Portal (TEP)
Desktop Client HTTP Browser
Tivoli Enterprise Console (TMR/TEC)
REMOTE Tivoli Enterprise Monitoring Server (TEMS)
TCP/1918
TCP/5529
TCP/1918
TCP/6014
TEMS DB
Tivoli Enterprise Monitoring Agents (TEMA)
TCP/1920
TCP/15001
TCP/1918
Tivoli Enterprise Portal Server (TEPS)
HUB Tivoli Enterprise Monitoring Server (TEMS)
TCP/6014
Historical Data Flow
TEPS DB
TCP/1918
TCP/1918
TEMS DB
TCP/1918
Tivoli Enterprise Monitoring Agents (TEMA)
TCP/1918
REMOTE Tivoli Enterprise Monitoring Server (TEMS)
TCP/ODBC
TCP/6014
Warehouse Proxy (WPA)
TCP/1918
TEMS DB
Tivoli Data Warehouse DB (TDW)
TCP/ODBC
Historical Data Flow
Summarization & Pruning Agent
TDW DB
Calculated port: 6014
KDC_FAMILIES=IP.PIPE COUNT:1 PORT:1918 IP use:n SNA use:n IP.SPIPE use:n
COUNT:1 means only 1918+(1*4096)=6014 will be used, thus the proxy port is fixed

Figure 2-12 Advanced installation on more secure side

This scenario forces the TEMA to warehouse traffic through the firewall. The Warehouse Proxy agent and the Tivoli Data Warehouse repository are both located in the more secure zone. This design increases the complexity for the Warehouse Proxy agent but also increases the security of the warehouse data.

To open the proper ports so that the TEMA can warehouse the historical data through the firewall, the Warehouse Proxy agent must establish a well-known listening port. This well-known port is calculated through the KDC_FAMILIES mechanism.

Tip: The Warehouse Proxy agent calculated port is significant only when:

- The TEMAs are warehousing data directly to the Warehouse Proxy agent, instead of storing on the Remote TEMS.
- Firewall policies do not allow ODBC connections to be made from less secure to the more secure infrastructure, and the Warehouse Proxy agent must be located behind the firewall from the agents.
- The TEMA must go through a firewall to connect to the Warehouse Proxy agent.

When warehousing data in a large installation especially within the boundaries of firewalls, keep these tips handy:

- Accurately calculate the collection amount of historical data. Firewall traffic can increase excessively when Historical Data Collection is enabled.
- Only collect the critical attribute groups, being careful not to turn on unnecessary attribute groups.
- Historical data roll-up can be stored on the Remote TEMS. However, it is severely limited to 250 TEMAs per Remote TEMS.
- Windows has a limit of a maximum 2,000 sockets open simultaneously. Within a firewall environment, IP.PIPE is required. This limits the warehousing of historical data to only 1,500 TEMAs (500 sockets are reserved for internal processing) per IBM Tivoli Monitoring installation.

Important: If you are familiar with the functions of Tivoli Firewall Security Toolbox (TFST), IBM Tivoli Monitoring 6.1 firewall support currently does not provide all of the functions of the TFST, particularly to be able to start the connection from the secure site and proxying between multiple firewalls (to be able to use different ports between multiple firewalls). These functions are expected to be available with a post-GA fix pack for IBM Tivoli Monitoring 6.1.

2.2.6 Advanced huge installation: multiple TEMS processes

This advance deployment scenario illustrates the power, flexibility, and capabilities of IBM Tivoli Monitoring. This deployment strategy requires double the recommended hardware specifications but less physical hardware deployment. This deployment exposes the technical capacity of running multiple monitoring server (TEMS) processes on one physical system. It certifies the adaptability of IBM Tivoli Monitoring, but acknowledges the extreme complexity that can occur within an installation.

Exceptional planning and assessment must precede this implementation. It can be very easy to allow this strategy to become a maintenance dilemma.

This installation still requires multiple Hub TEMSs but leverages unused hardware capacity to run additional Remote TEMS processes (configured to listen on different ports) connecting to the separate Hub TEMS.

The installation consists of the following components:

- Tivoli Enterprise Monitoring Server
- Tivoli Enterprise Portal Server
- Tivoli Enterprise Portal
- Tivoli Warehouse Proxy agent
- Tivoli Data Warehouse
- Summarization and Pruning agent
- Tivoli Enterprise Console

This design strives to highlight the potential strategic deployment using multiple TEMS processes configured on different listening ports. It is architecturally similar to the large installation. However, there are multiple Remote TEMS processes running on one physical system.

Figure 2-13 on page 47 demonstrates a simple architecture to present the underlying theory. IBM Tivoli Monitoring can expand farther and run more than two kdsmain processes per system. This technique accomplishes a larger implementation with less physical hardware. To keep the diagram intelligible, the Remote TEMS have been stacked for brevity. Even though this strategy is similar to the large installation (the implementation is exactly the same), we do not recommend loading this IBM Tivoli Monitoring installation to its maximum throughput.

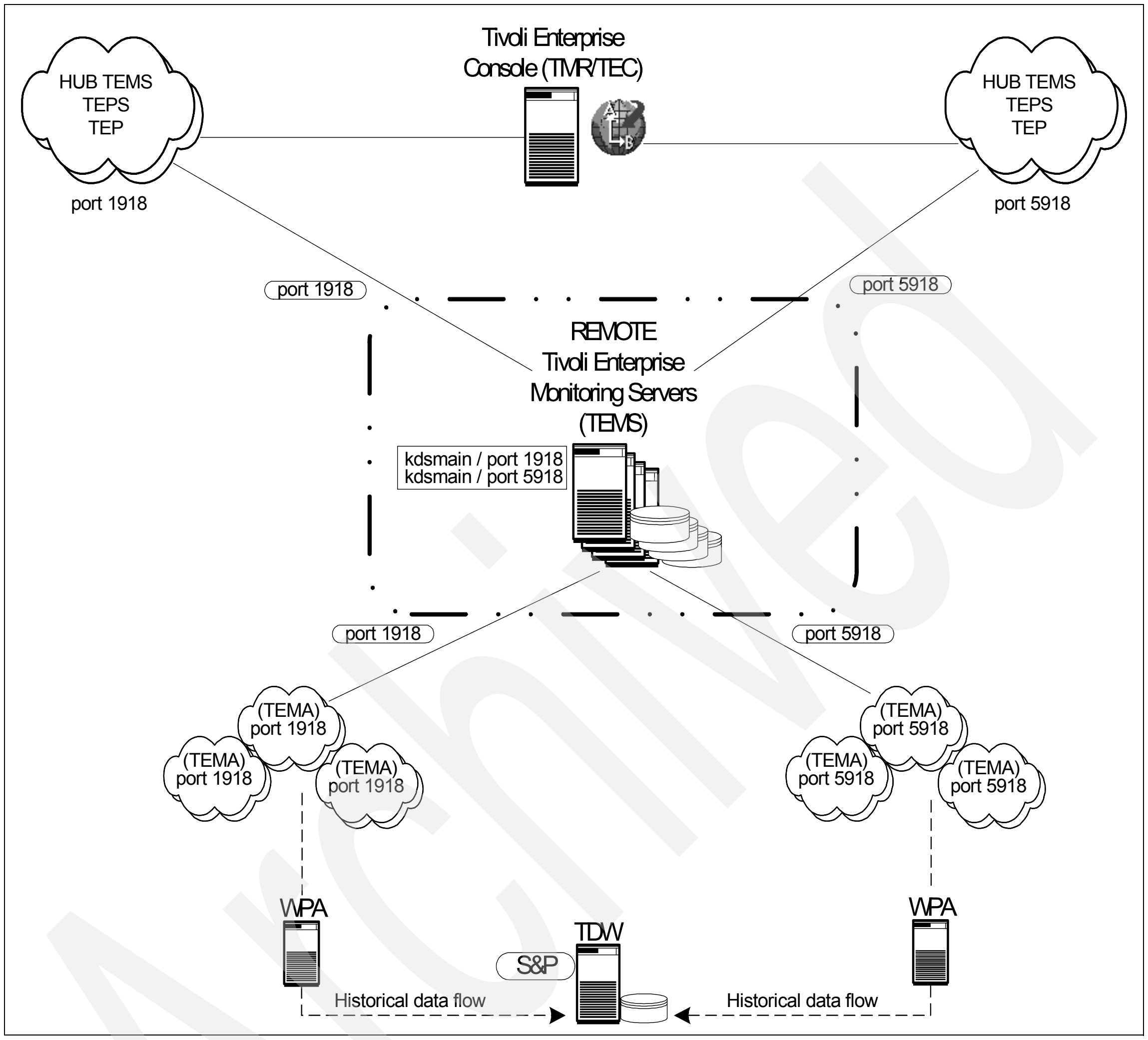


Figure 2-13 Large installation with multiple TEMS processes on single system

Essentially, this large installation is capable of having up to 10 Remote TEMS, each running two or more Tivoli Enterprise Monitoring Server processes within one system boundary.

IBM Tivoli Monitoring has a software limitation that permits only a single TEMS process to listen on the well-known port (default port 1918). To achieve this design, any additional TEMS processes on the exact system must be configured to use an unused port. This process essentially doubles the IBM Tivoli Monitoring capacity without additional hardware. We advise running only one Hub TEMS per physical system. Although supported, multiple TEMS processes on a single system all configured as (*HUB) is not suggested.

To accomplish this installation, separate IBM Tivoli Monitoring install directories are necessary. These installations will be completely separate despite the binaries existing on the same physical system. Follow the normal installation procedures for a single large installation environment. The only necessary installation procedure change is the configuration of the well-known listening port for each running TEMS process. For Historical Data Collection, it is exactly the same as a huge installation: one Tivoli Data Warehouse repository with multiple Warehouse Proxy agents.

Note: A Warehouse Proxy agent is required for each separate TEMS process instance. For example, TEMS process instances running on three different ports will require three Warehouse Proxy agents.

One thought to keep in mind: These multiple TEMS processes are still logically independent installations that each require separate maintenance. The infrastructure systems will have distinct CANDLEHOME installation directories. The IBM Tivoli Monitoring code on every system will require maintenance to every separate installation directory.

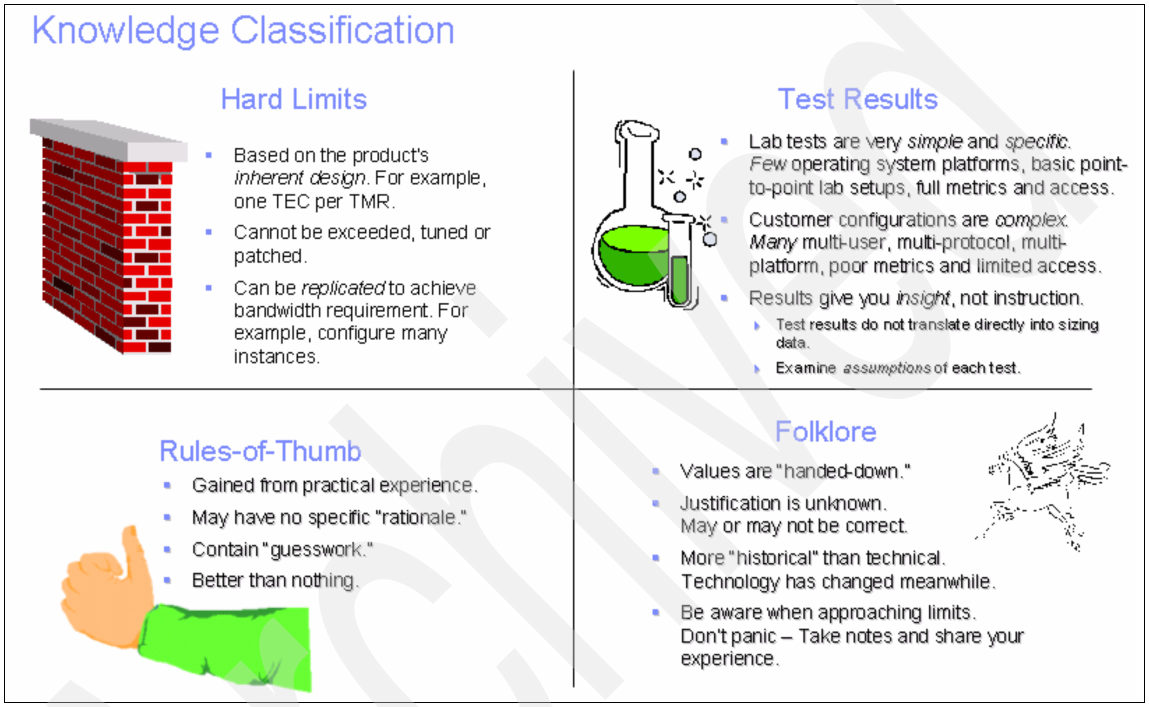
Attention: Running multiple TEMSs on a single system is technically supported by IBM Tivoli Monitoring. However, all alternatives should be given careful consideration. The total cost of ownership to maintaining this complex environment can be higher than the cost of additional hardware capacity.

2.3 Scalability

A distributed networking infrastructure inherits scalable characteristics by design. After all, a distributed system is built to expand and shrink through the increment and decrement in hardware capacity. It should be stated that scalability is not the same as performance tuning. Performance tuning deals with increasing output from current capacity without adding additional resources.

No single analysis of scalability and performance can determine the absolute hard limits of a distributed product. A distributed system in theory should extend to infinity. However, as distributed systems increase in scalability, performance loss may increase to an unsustainable boundary. IBM Tivoli Monitoring follows the basic scalable characteristic in this design. Adding hardware capacity in the form of remote TEMS distributes the load and allows more connected agents. This methodology represents a foundation that is built upon using the actual calculated values from the physical environment. Note that IBM Tivoli Monitoring constitutes vast improvements in scalability and performance from OMEGAMON XE, highlighting the union between the mature code of Candle Corporation and the enterprise qualification of IBM Tivoli Software.

Figure 2-14 depicts a great universal example of many unique sources of information pertaining to scalability and performance metrics. It exposes the issues related to scalability and performance expectations.



Knowledge Classification

Hard Limits

- Based on the product's *inherent design*. For example, one TEC per TMR.
- Cannot be exceeded, tuned or patched.
- Can be *replicated* to achieve bandwidth requirement. For example, configure many instances.

Test Results

- Lab tests are very *simple* and *specific*. *Few* operating system platforms, basic point-to-point lab setups, full metrics and access.
- Customer configurations are *complex*. *Many* multi-user, multi-protocol, multi-platform, poor metrics and limited access.
- Results give you *insight*, not instruction.
  - Test results do not translate directly into sizing data.
  - Examine *assumptions* of each test.

Rules-of-Thumb

- Gained from practical experience.
- May have no specific "rationale."
- Contain "guesswork."
- Better than nothing.

Folklore

- Values are "handed-down."
- Justification is unknown. May or may not be correct.
- More "historical" than technical. Technology has changed meanwhile.
- Be aware when approaching limits. Don't panic – Take notes and share your experience.

Figure 2-14 Universal sources of scalability and performance numbers

For example:

- User Guide says "unlimited"
- Support said "no more than n …"
- Development said "n*3 …"
- Early Support Program said "n^3 …"
- Services said "n/3 ..."
- Performance said "we didn't test that."

A decision must be chosen carefully because different sources have their own reasons for providing sizing metrics.

For IBM Tivoli Monitoring, analysis of all of these sources, including an in-depth knowledge of the monitoring environment, assists in scaling the installation properly. Understanding the limitations of IBM Tivoli Monitoring and strategically working through them will facilitate obtainable goals.

From a scalability standpoint, the TEMS plays the key role. As the architect of an IBM Tivoli Monitoring implementation, consider the following factors:

- Number of physical hosts and platforms types included
- Number and type of applications and operating systems per host
- Geographical topology of the environment, particularly in relation to where the managed systems shall reside
- Estimated number of events generated, thresholds that will be deployed, or both
- The degree of automation that is required or planned – both reflex and workflow
- Estimated number of TEP users and the expected type of usage (heavy reporting, frequent real time updates, and so forth)
- Network topology and firewall considerations

The information generated from these points above can then be combined with the scalability guidelines that have been established for the initial release of IBM Tivoli Monitoring.

The following scalability metrics are from verification testing performed on IBM Tivoli Monitoring (GA). These numbers represent actual test synopsis validation. These numbers are not definitive declarations of scalability and performance. This data displays achievable goals that have been proven in a test/development environment. All IBM Tivoli Monitoring installations are unique and require surveillance during the deployment.

Table 2-3 classifies the extensive metrics for IBM Tivoli Monitoring. These metrics measure the apex for the IBM Tivoli Monitoring components with respect to load quantity. Each metric represents one installation instance.

Table 2-3 Extensive metrics

IBM Tivoli Monitoring component	**Verified metric**
Remote TEMS	15 (Windows and UNIX)
Managed Systems	5,000
Managed Systems per Remote TEMS	500
Heartbeating agents per TEMS	500

IBM Tivoli Monitoring component	Verified metric
Simultaneous agent startup/logins to a TEMS	1,000
Agents storing historical data at Remote TEMS	250
Consoles per TEPS	50
Total situations	1,500 (30/agent)

Important: These metric values do not represent actual hard limits in IBM Tivoli Monitoring. These numbers are derived from what was actually tested, not necessarily product limitation.

The Tivoli Data Warehouse scalability and metrics are beyond the scope of this chapter. For detailed information about performance and planning guidance, refer to Chapter 4, “Historical summarized data” on page 185.

2.4 Agent deployment architecture

There are several techniques for installing the Tivoli Enterprise Management Agents. This section summarizes three common practices that can be employed to install managed systems with an installation.

All three scenarios include the positives and negatives of an established solution. Proper assessment of the physical environment is part of the decision of which solution best fits.

Tips to keep in mind:

- Total number of physical systems and the total amount of agents deployed to each of those systems
- Network bandwidth and latency between TEMS and TEMA
- Size of the IBM Tivoli Monitoring installation
- Connectivity to the managed systems

2.4.1 IBM Tivoli Monitoring built-in deployment controller

IBM Tivoli Monitoring offers an easy, efficient deployment mechanism to push Operating System Agents and Applications Agents to remote systems. This mechanism also offers agent upgradability. IBM Tivoli Monitoring provides a powerful built-in tool for intelligent agent upgrades via the GUI or command line.

Figure 2-15 shows the architecture of IBM Tivoli Monitoring agent components. The functionality of the agent components is divided among the TEPS, TEMS, and OS Agent, respectively.

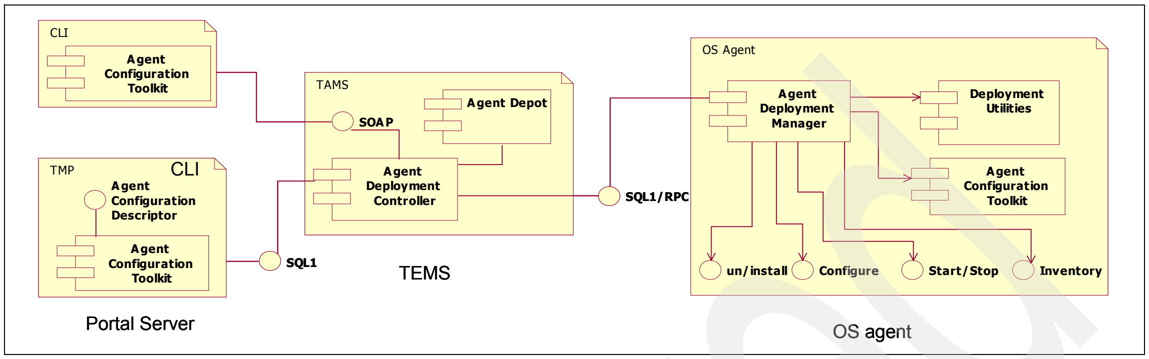


Figure 2-15 Agent deployment architecture

IBM Tivoli Monitoring OS Agents, implemented as a DLL, can handle agent deployment activities at the agent end.

The Agent Depot is an installation directory on the monitoring server from which you deploy agents and maintenance packages across your environment. The Agent Depot must reside on a local TEMS or on a remote file system configured as its depot home directory. Before you can deploy any agents from a monitoring server, you must first populate the Agent Depot with bundles. A *bundle* is the agent installation image and any prerequisites.

Agents can be loaded into the Agent Depot at install time. Installer on Windows and UNIX has a "populate depot" option.

Note: No transfer of packages from one TEMS to another is provided.

Each agent bundle in the Agent Depot can be determined by its product ID and platform characteristics. The Agent Depot can also contain MDL files and scripts used in the deployment of the Universal Agent. You can customize the Agent Depot based on the types of bundles that you want to deploy and manage from that monitoring server.

The deployment controller, a service on the Management Server, acts as the driver for the deployment. The deployment controller queries the Agent Depot contents and transfers agent bundles using Remote Procedure Calls (RPC). All other tasks are initiated by making SQL1 calls. Agent deployment requests are made using SQL1 calls to a Management Server. The deployment controller provides the ability to initiate deployment commands from a SQL1 interface.

Notes:

- To allow remote deployment, the target system must support (and be configured for) at least one of these protocols:
  - SMB (server message block)
  - SSH (secure shell protocol)
  - REXEC (remote execution protocol)
  - RSH (remote shell)

  By default, the deployment controller attempts each of these protocols until a connection is successfully established on one of them.
- Remote Procedure Call (RPC) is a protocol that one program can use to request a service from a program located in another computer in a network without having to understand network details. (A procedure call is also sometimes known as a function call or a subroutine call.)
- SQL1 is the SQL implementation based on the ANSI-1989 SQL1 standard.

Deployment controller commands can be targeted to a specific system or to a managed system list. The deployment controller manages the interaction with the management agent (OS Agent); it manages receiving and aggregating results from multiple targets and provides forwarding of requests to the appropriate TEMS as well as queuing of requests for scalability. The following processes can be initiated: install, uninstall, and upgrade.

Note: Deployment requests are asynchronous; when a request is received it is queued up for processing.

Agents vary greatly in how they are configured depending on the agent type and the OS platform. The Agent Configuration Toolkit collects and transfers configuration data. It provides a set of utilities that enable the agent deployment to configure agents. The Agent Configuration Toolkit and the deployment controller communicate via SOAP (Simple Object Access Protocol).

SOAP is a way for a program running in one kind of operating system (such as Windows 2000) to communicate with a program in the same or another kind of an operating system (such as Linux) by using the World Wide Web's Hypertext Transfer Protocol (HTTP) and its Extensible Markup Language (XML) as the mechanisms for information exchange. Because Web protocols are installed and available for use by all major operating system platforms, HTTP and XML provide a ready solution to the problem of how programs running under different operating systems in a network can communicate with each other. SOAP specifies exactly how to encode an HTTP header and an XML file so that a program in one computer can call a program in another computer and pass it information. It also specifies how the called program can return a response.

An advantage of SOAP is that program calls are much more likely to get through firewall servers that screen out requests other than those for known applications (through the designated port mechanism). HTTP requests are usually allowed through firewalls, so programs using SOAP to communicate can be sure that they can communicate with programs anywhere.

2.4.2 Tivoli Configuration Manager V4.2

Most IBM Tivoli Software customers already have an investment in Tivoli Management Framework V4.1.1 and IBM Tivoli Configuration Manager V4.2. IBM Tivoli Monitoring Agents can be deployed using IBM Tivoli Configuration Manager V4.2 as the delivery mechanism.

It is cost-effective to leverage IBM Tivoli Configuration Manager V4.2 as a solution to deliver IBM Tivoli Monitoring Agents. IBM Tivoli Configuration Manager V4.2 is robust and designed for large-volume software pushes, which dominates over IBM Tivoli Monitoring Deployment Controller.

Refer to Chapter 11, “Agent deployment using IBM Tivoli Configuration Manager Software Distribution” on page 633 for scenarios about installing TEMAs using IBM Tivoli Configuration Manager V4.2.

2.4.3 Operating system image deployment

It is possible to *manually* extract the package files to generate a customized image to transfer to an operating system image for replication. The technique is similar to the Tivoli Configuration Manager V4.2 method. The only difference is that software distribution is not used to push the install packages.

The install packages are built and then transferred to a pristine operating system image that gets deployed to many systems using a third-party method.

After the operating system is built from the image, the silent install can be leveraged to install the product binaries via the standard silent install mechanism.

Operating system imaging is beyond the scope of this book. This is an alternate method that can be employed and is another recommended deployment solution.

Refer to the software package definitions in Appendix A, “TEMA Software Package Definition examples” on page 753 for further details about discovering the individual product packages for all IBM Tivoli Monitoring components.

3

IBM Tivoli Monitoring 6.1 installation and first customization

In this chapter, we discuss detailed steps and best practices to implement IBM Tivoli Monitoring 6.1 using two different scenarios: Windows TEMS and UNIX TEMS. We offer best practices guidelines in terms of machine sizing and configurations.

Note: This chapter discusses IBM Tivoli Monitoring 6.1 implementation from scratch. If your requirement is to migrate from Tivoli Distributed Monitoring 3.7 or OMEGAMON XE, refer to Chapter 5, “Tivoli Distributed Monitoring V3.7 upgrade” on page 239 and Chapter 7, “OMEGAMON XE Upgrade” on page 407, respectively.

In 2.2, “IBM Tivoli Monitoring deployment scenarios” on page 23, we discussed several sizes of IBM Tivoli Monitoring 6.1 installations. The instructions given in this chapter can be applied to all of these installations, although additional considerations are discussed for huge-size installations in 2.2.4, “Huge installation (greater than 4000 agents)” on page 31.

This chapter discusses the following topics:

- Lab environment
- Installing IBM Tivoli Monitoring 6.1
- Uninstalling IBM Tivoli Monitoring 6.1

3.1 Lab environment

The following section describes the software and hardware components used during the implementation of IBM Tivoli Monitoring 6.1 in our lab environment. It also outlines the architecture used to build that environment.

To simulate real-time environments as much as possible, we set up two different scenarios: Windows-based TEMS and UNIX-based TEMS.

- 3.2.5, "Installing and configuring the scenario 1 environment" on page 69 covers a typical IBM Tivoli Monitoring 6.1 implementation with two Windows TEMS servers with second server used for Hot Standby.
- 3.2.16, "Installing and configuring the scenario 2 environment" on page 169 discusses an implementation with two UNIX TEMS servers, with a second server again used for Hot Standby.

Apart from the TEMS servers, all IBM Tivoli Monitoring 6.1 components remain the same in both scenarios. You can use one of these scenarios depending on the TEMS platform of your choice.

If you install your TEMS server on one platform (such as Windows), then decide to migrate it to another platform (such as UNIX), you can follow the instructions given in 3.2.17, "Replacing a Hub TEMS server with a new one" on page 176.

Notes:

- Both of these configurations can be achieved with one TEMS server instead of two, if you do not plan to use the Hot Standby functionality. For fault tolerance reasons, we recommend that you use the Hot Standby function.
- It is also possible to use one Windows and one UNIX TEMS server, as Hot Standby functionality also works between Windows and UNIX servers. We tested this scenario successfully for this book.

3.1.1 Hardware and software configuration

Table 3-1 shows the hardware and software configuration of our lab environment.

Table 3-1 Lab hardware and software configuration

Server	OS	CPU	Memory	Hard disk	Main components	Specific applications
amsterdam	W2K/SP4	P4 3Ghz	514 MB	32 GB	TEMA	
berlin	W2K/SP4	P4 3Ghz	2 GB	32 GB	TEPS	DB2 8.2
cairo	W2K3	Xeon™ 3Ghz	3.5 GB	32 GB	Hub TEMS	
copenhagen	W2K/SP4	P4 1.8Ghz	1 GB	37 GB	Remote TEMS	
as20	AS/400®	E Series G170	512 MB	68 GB	TEMA	
izmir	W2K/SP4	P4 1.8Ghz	260 MB	22 GB	WPA & SPA	DB2 8.2
lizbon	W2K/SP4	P4 1.8Ghz	391 MB	27 GB	TEMA	
london	W2K/SP4	P4 3Ghz	512 MB	74 GB	TEP	
dakar	W2K/SP4	P4 1.8Ghz	260 MB	27.9 GB	TEMA	Exchange Server 2000
istanbul	AIX 5.3.0	F80 RS6K	1 GB	24 GB	Event synchronization	TEC
madrid	AIX 5.3.0	F80 RS6K	1 GB	36 GB	Event synchronization	
milan	AIX 5.3.0	F80 RS6K	2 GB	222 GB	Event synchronization	
ankara	RHEL4U1	P3 900Mhz	1 GB	37 GB	TEMA	
edinburg	RHEL4U1	P4 1.8Ghz	512 MB	40 GB	Remote TEMS	
oslo	SLES9	P4 1.8Gz	1 GB	40 GB	TEMA	

3.1.2 Lab architecture

The following sections describe the architecture on the two different scenarios.

Lab architecture of scenario 1

Figure 3-1 shows the first scenario with two Windows Hub TEMS servers and two Remote TEMS servers, one UNIX server, and one Windows server.

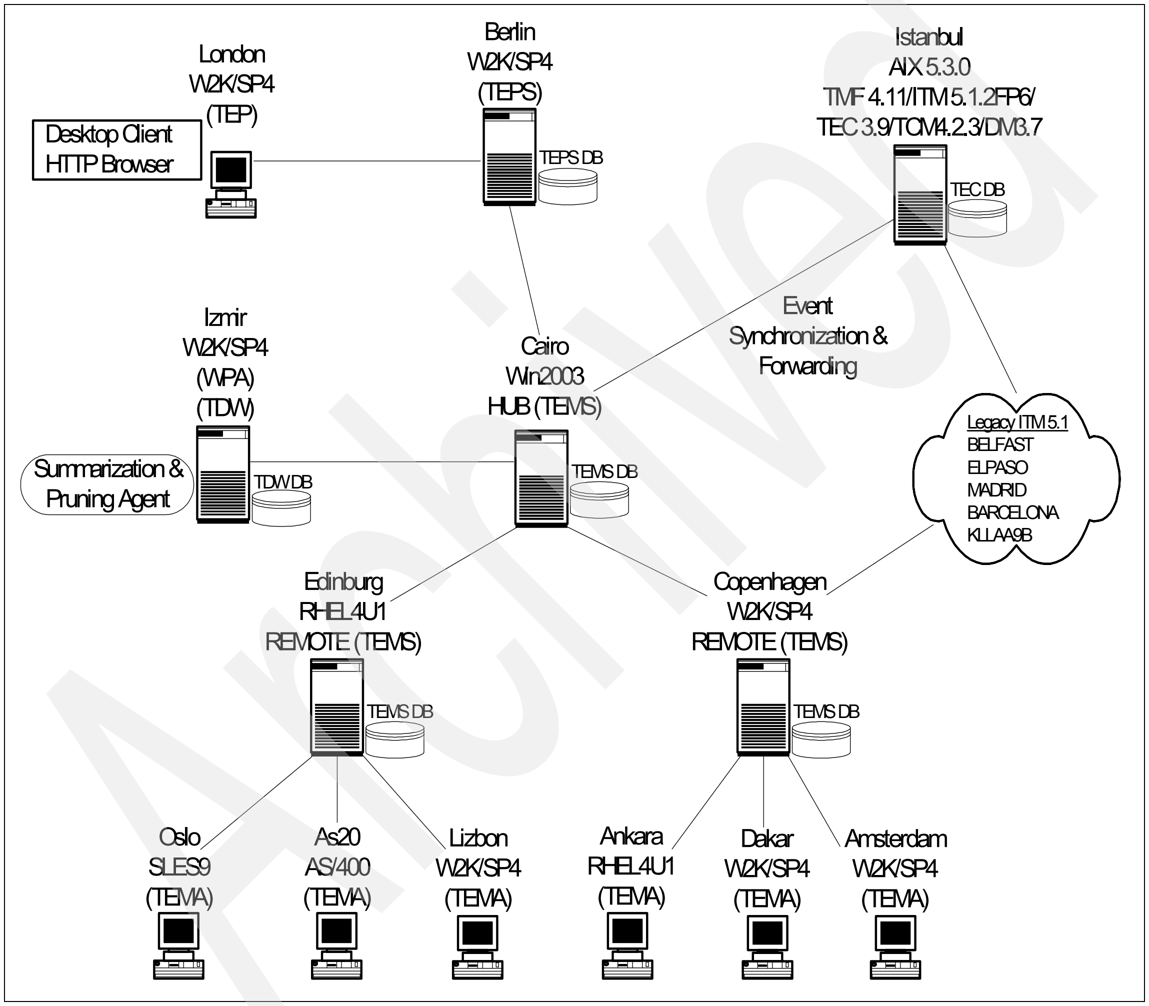


Figure 3-1 Lab architecture or a large-scale enterprise, scenario 1

Note: For simplicity, the Hot Standby node is not shown in the topology diagram. In a large installation, it is strongly recommended that you implement the Hot Standby node.

Lab architecture scenario 2

Figure 3-2 shows the second scenario with two AIX Hub TEMS servers and the same configuration for the rest of the components as scenario 1.

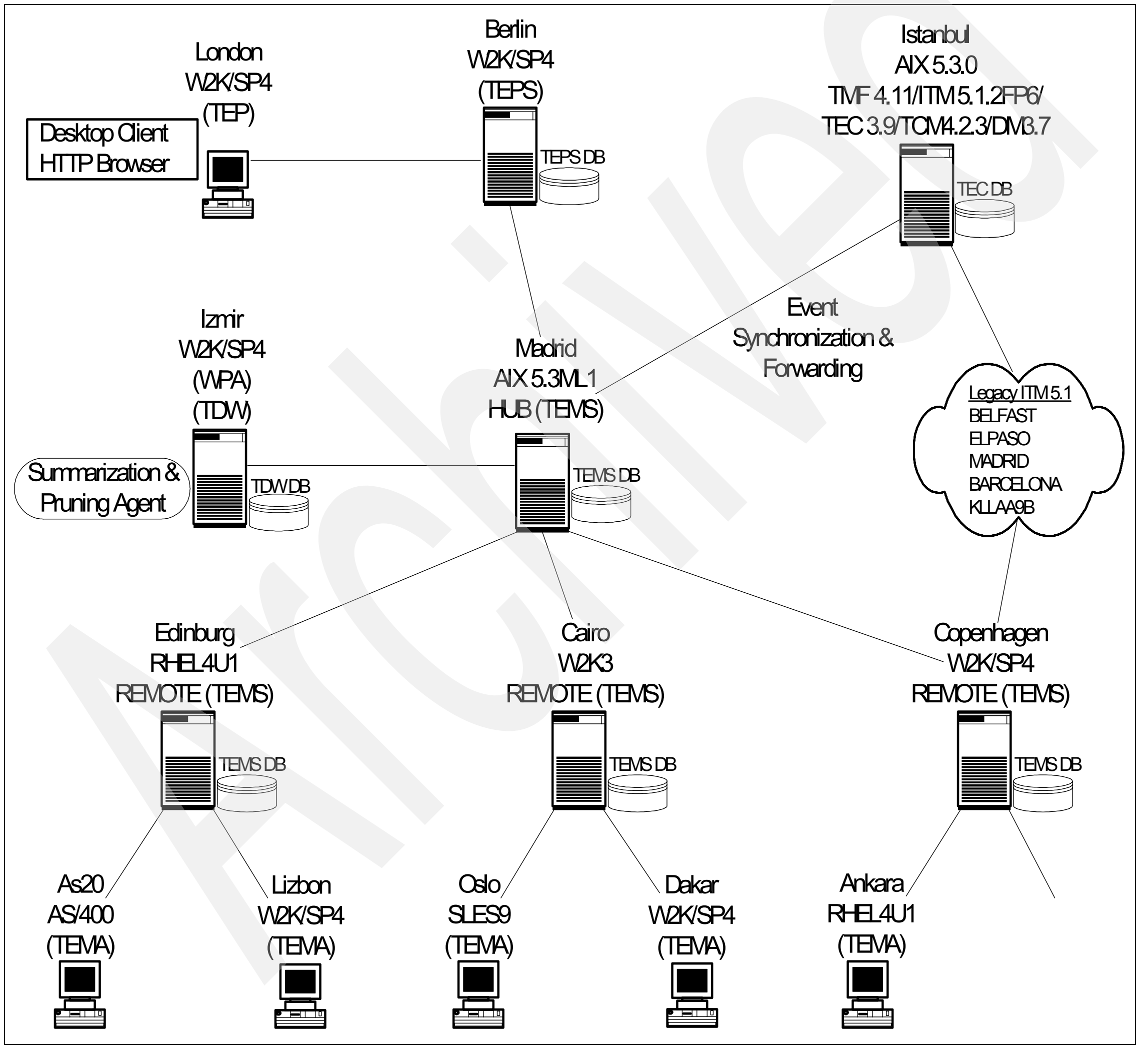


Figure 3-2 Lab architecture for a large-scale enterprise, scenario 2

This architecture is scalable and can be extended to contain more than 4000 agents. Refer to 2.2.4, “Huge installation (greater than 4000 agents)” on page 31 for more details about extending this architecture to handle more than 4000 agents.

To build the second scenario and reconfigure cairo as a Remote TEMS, we performed the steps described in 3.2.17, “Replacing a Hub TEMS server with a new one” on page 176.

3.2 Installing IBM Tivoli Monitoring 6.1

This section describes step-by-step the installation process of the different IBM Tivoli Monitoring 6.1 components, showing examples for both GUI and command line interface (CLI) installation.

Table 3-2 provides an overview of the steps required to fully install and deploy an IBM Tivoli Monitoring 6.1 environment.

Table 3-2 Installation steps

Steps	References
Planning the installation	“Planning the installation” on page 62
Install the Tivoli Enterprise Monitoring Server	“Installing and configuring a Hub TEMS on a Windows server” on page 69 and “Installing a Hub TEMS on a UNIX server” on page 169
Install the Tivoli Enterprise Remote Monitoring Server	“Installing a Remote TEMS on a Windows and UNIX server” on page 82
Install the Tivoli Enterprise Portal Server	“Tivoli Enterprise Portal Server - TEPS” on page 87
Install Tivoli Management Agent	“Tivoli Enterprise Monitoring Agent” on page 95
Install the portal desktop client on any system where you want to use it	“Tivoli Enterprise Portal (TEP)” on page 126
Install Warehouse Proxy	“Installing the Warehouse Proxy agent” on page 132
Install TEC event synchronization	“Event synchronization installation” on page 147

3.2.1 Planning the installation

This section outlines the information that you need to have ready before starting the installation.

We discuss the following topics:

- ► Expertise required
- ► Naming the monitoring server
- ► Creating an IBM Tivoli account on UNIX servers
- ► Import the images
- ► Host name for TCP/IP network services
- ► Use of fully qualified path names
- ► File descriptor (maxfiles) limit
- ► Hardware and software prerequisites

Expertise required

Installing IBM Tivoli Monitoring 6.1 requires expertise in several areas. In this section, we describe some of the general expertise required to perform the tasks listed above. We take each phase of the installation down to the operating system level. In most cases, it is not necessary for one person to have all of the expertise, but if the person designated as the Tivoli administrator possesses some knowledge of these different products, it will be easier to deploy and maintain the product.

It is not unusual to have different people with expertise in these technologies working together. The workload of the Tivoli administrator will be significantly more than the workload of, for example, the DB2 administrator in most cases.

Database administrator

The database administrator (DBA) must possess an understanding of how databases work. A DBA, if there is one in the organization, needs to know how to perform most of the work to be done with the chosen database. A thorough knowledge of the RDBMS is not needed in most instances, but will greatly enhance the IBM Tivoli Monitoring 6.1 experience.

Operating systems administrator

The UNIX administrator should have advanced knowledge of how to administer a UNIX server. The UNIX administrator will be called on at times to upgrade the operating system software and any other software that is resident on the server. The UNIX administrator must also be able to add and delete users and be able to change permissions on directories. There is also a need to do some debugging at the operating system level and know about TCP/IP, networks, Domain Name System (DNS), host files, file systems, cron jobs, ports, adding additional space for growing processes, and any other assignments that a UNIX administrator would do in the normal course of business.

The Microsoft® Windows administrator should have the advanced knowledge of administering Microsoft Windows computers, both server classes, such as Microsoft Windows 2000 Servers or Microsoft Windows 2003 Servers, and workstation classes such as Microsoft Windows XP. This administrator will be called on to keep the operating system software updated and secure, and must be able to give certain security rights to the Tivoli administrator that will enable the software to operate normally.

3.2.2 Define the architecture

Chapter 2, “Architecture and planning” on page 15 gives a complete and detailed description of the best practices to define an architecture that fits into your organization. We set up our lab architecture based on those best practices. The two scenarios resulting from those suggestions are described in Figure 3-1 on page 59 and Figure 3-2 on page 60.

Note: Although IBM Tivoli Monitoring 6.1 can be set up using two hubs with different platforms (Windows and UNIX, for example), if you are planning to implement the Hot Standby feature, it is advisable (but not mandatory) to have them on the same platform.

3.2.3 Creating a deployment plan

A deployment plan is essential for creating and installing an IBM Tivoli Monitoring 6.1 environment.

You must gather at least the following information before installing any software:

- Base hardware and software requirements for IBM Tivoli Monitoring 6.1. This information is provided in *IBM Tivoli Monitoring Installation and Setup Guide*, GC32-9407.
  - Whether the computer systems in your distributed network can support this new software, or these systems can be upgraded to meet your business needs, or whether new systems need to be obtained.
  - Which IBM Tivoli Monitoring 6.1 components to install on which computer systems in your distributed network to support your business needs and whether they have additional third-party software requirements. This information is provided in 2.1.1, “Platform support matrix for IBM Tivoli Monitoring” on page 22.

- For each system where you plan to install components of IBM Tivoli Monitoring 6.1, gather the following information:
  - Name of the monitoring server you are installing or that the agent will connect to
  - Operating system
  - Available memory and available disk space
  - Host name of the system where the product (a monitoring server or one instance of an agent) will execute.
  - Whether the monitoring server being installed will be configured as a hub or remote monitoring server
  - Hub host name
  - Port number

Naming the monitoring server

In general, the names should be short but meaningful within the environment. Use the following mandatory guidelines when selecting the names of monitoring servers:

- Each name must be unique. One name cannot match another monitoring server name for its entire length.
- Each name must begin with an alpha character. No blanks or special characters (.$#@.) can be used.
- Each name must be between two and 32 characters in length.
- Management server naming is case-sensitive on all platforms.

Table 3-3 and Table 3-4 on page 65 describe the various TEMS configuration on our two environments.

Table 3-3 Scenario 1 lab TEMS description

Monitoring server	Host name	Architecture	Description
HUB_HELSINKI	helsinki	W2K3	TEMS Hub
HUB_CAIRO	cairo	W2K3	TEMS Hub
REMOTE_COPENHAGEN	copenhagen	W2K	TEMS Remote
REMOTE_EDINBURG	edinburg	Redhat 4	TEMS Remote

Table 3-4 Scenario 2 lab TEMS description

Monitoring server	Host name	Architecture	Description
HUB_MADRID	madrid	AIX F80	Hub TEMS
HUB_MILAN	milan	AIX F80	Hub TEMS
REMOTE_CAIRO	CAIRO	W2K3	Remote TEMS
REMOTE_COPENHAGEN	copenhagen	W2K	TEMS Remote
REMOTE_EDINBURG	edinburg	Redhat 4	TEMS Remote

Creating an IBM Tivoli account on UNIX servers

We created an IBM Tivoli account for installing and maintaining the installation directory. For best performance, follow these guidelines:

- You can use any valid name. You can install the IBM Tivoli Monitoring software as the root user on UNIX, but you do not have to. If you do not install IBM Tivoli Monitoring 6.1 as root, you must use the following procedure to create the user and correctly set the permission. We created a user called *itmuser* in *itmusers* group. IBM recommends using the Korn shell for your IBM Tivoli account; however, you can use any shell that is shipped with the UNIX operating system.

  a. Create the itmusers group using the following procedures.

     For Linux, Solaris, and HP-UX computers, run the following command:

     ```
     groupadd itmusers
     ```

     For an AIX computer, run the following command:

     ```
     mkgroup itmusers
     ```

  b. Create the *itmuser* user belonging to *itmusers* group; itmusers will be the primary itmuser group.

     For AIX, Solaris and Linux computers run the following command to create the *itmuser* account:

     ```
     useradd -g itmusers -s /usr/bin/ksh itmuser
     ```

- The same user should install all components.
- If you are using NFS or a local file system, you should establish your installation directory according to the guidelines used in your environment.

When the user is properly created, use the following procedure to set the permissions:

  a. Set the CANDLEHOME directory. You must use the itmuser user profile.

     ```
     export CANDLEHOME=/opt/IBM/ITM
     ```

b. Run the following command to ensure that the CANDLEHOME environment variable correctly identifies IBM Tivoli Monitoring installation directory:

```
echo $CANDLEHOME (default is /opt/IBM/ITM)
```

Attention: Running the following steps in the wrong directory can change the permissions on every file in every file system on the computer.

c. Change to the directory returned by the previous step:

```
cd $CANDLEHOME
```

d. Run the following command to ensure that you are in the correct directory:

```
pwd
```

e. Run the following commands:

```
chgrp itmusers .
chgrp -R itmusers .
chmod o-rwx .
chmod -R o-rwx .
```

Important: If you did this operation after the agent installation, run the following command to change the ownership of additional agent files:

```
bin/SetPerm
```

Select **All of the above** to set the proper permission on all installed agents.

Import the images

Import the IBM Tivoli Monitoring 6.1 images to the server where you will perform the installation.

You can create a separate file system where you will download the images and install IBM Tivoli Monitoring 6.1. To install on a Windows system, create a drive distinct from the C: drive (or whatever drive the operating system resides on).

Host name for TCP/IP network services

TCP/IP network services such as NIS, DNS, and the /etc/hosts file should be configured to return the fully qualified host name (*hostname*.ibm.com, for example). Define the fully qualified host name after the dotted decimal host address value and before the short host name in the /etc/hosts.

Execute the following command line from the TEMS:

```
nslookup hostname
```

In this example, `hostname` is host name of the servers in the IBM Tivoli Monitoring 6.1 environment (for example, the second Hub TEMS, the Remote TEMS, the TEPS, and so on.). After the command is performed successfully on those servers, proceed with the reverse lookup executing the following command:

```
nslookup -querytype=PTR
```

At the prompt, enter the IP address of the previously tested server. Fix any inconsistencies by contacting your System or Network Administrator before proceeding to the next steps.

Use of fully qualified path names

Because of the wide variety of UNIX operating systems and possible user environments, use *fully qualified* path names when entering a directory during the installation process (no pattern-matching characters). IBM scripts use the Korn shell—when a new process or shell is invoked, use of symbolic links, environmental variables, or aliases can potentially cause unexpected results.

File descriptor (maxfiles) limit

The monitoring server requires a minimum of 256 file descriptors (maxfiles) for the operating system. For the monitoring server to function properly, we set the maximum file descriptor (MAX_FILES parameter of the configurable kernel parameter) to 256.

To determine the number of per-process file descriptors (maxfiles), run one of the following commands:

- `sysdef | grep maxfiles`
- `ulimit -a`

For AIX computers, run the following command:

```
ulimit -d
```

The `-d` option specifies the size of the data area in kilobytes. If the settings returned are less than 256 MB, increase the maxfiles limit to 256 MB.

If 256 MB is not sufficient (for example, as evidenced by the malloc failures in the monitoring server log file), contact IBM Software Support regarding a memory upgrade patch. This patch enables you to use multiple user segments of 256 MB. This patch must be applied to the KDSMAIN module at every product or maintenance installation.

Hardware and software prerequisites

All information regarding software and hardware prerequisites can be found in the *IBM Tivoli Monitoring Installation and Setup Guide*, GC32-9407. Read this document carefully to check whether your environment complies with the IBM Monitoring 6.1 prerequisites.

3.2.4 Backup strategies

This section describes several of the backup strategies that should be deployed when using IBM Tivoli Monitoring 6.1. Without a good backup strategy, the enterprise can be vulnerable to outages of indeterminate lengths of time.

Tivoli backups

If the IBM Tivoli Monitoring 6.1 is being installed in an existing Tivoli server, we suggest that you back up the Tivoli database just in case, even though IBM Tivoli Monitoring 6.1 does not update the Tivoli Framework database. There are two ways to back up the Tivoli server and managed nodes. The **`wbkupdb`** command is available to back up the pertinent files in the $DBDIR directory. This does not back up custom scripts or anything outside of $DBDIR, but it is sufficient to restore a Tivoli server to running state if there is corruption in the database.

System-level backups

The other method is to do a system-level backup, or back up everything under the ../Tivoli directory. This captures all scripts that were built for tasks and other custom scripts in that environment.

If you are installing IBM Tivoli Monitoring 6.1 in a TMR, make sure that you have a "clean" system before backing up. Use the **`wchkdb`** command with the appropriate parameters:

-ux	For interconnected Tivoli regions
-u	For all managed nodes
-ut	For just the Tivoli server

A "clean" **`wchkdb`** command result allows for better backup and restore capabilities.

One of the steps that is almost always forgotten is to check the backups for validity. Too often there are backups that are not validated, and they might not be good candidates when needed for restore purposes. A good plan is to have a machine that you can use to restore the backup, then use a set of tests to make sure that the backup is valid. If it is not, debug the problem to make sure that you can back up successfully. One debugging tip is to back up individual managed nodes and not the whole Tivoli region at a time. If there is a failure, you can see which managed node has the failure and further debug just that node.

3.2.5 Installing and configuring the scenario 1 environment

This section describes the different IBM Tivoli Monitoring 6.1 components' installation and configuration on an Windows Hub TEMS environment.

Installing and configuring a Hub TEMS on a Windows server

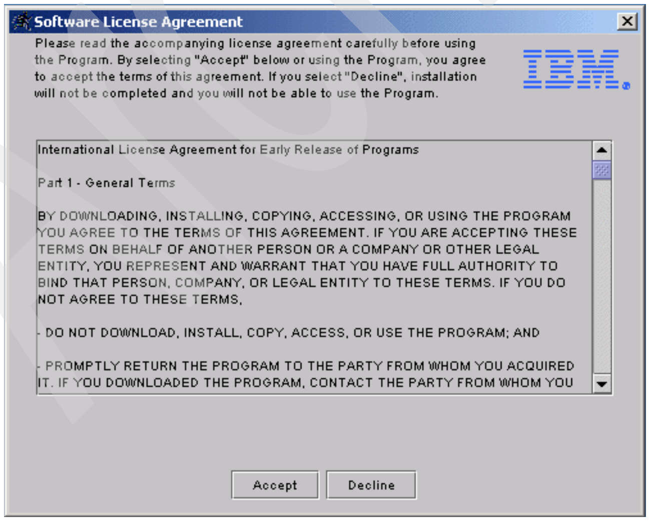
The following sections provide detailed information about installing a Hub TEMS on a Windows server and performing the initial configuration.

Use the following steps to install the hub monitoring server on a Windows computer:

1. Launch the installation wizard by double-clicking the **setup.exe** file on the installation media.
2. Click **Next** on the welcome window.

> **Note:** If you are running Windows 2003 or Windows XP and have security set to check the software publisher of applications, you might receive an error stating that the setup.exe file is from an unknown publisher. Click **Run** to disregard this error message and continue.

3. Click **Accept** to accept the license agreement, as shown in Figure 3-3.


Software License Agreement

Please read the accompanying license agreement carefully before using the Program. By selecting "Accept" below or using the Program, you agree to accept the terms of this agreement. If you select "Decline", installation will not be completed and you will not be able to use the Program.

International License Agreement for Early Release of Programs

Part 1 - General Terms

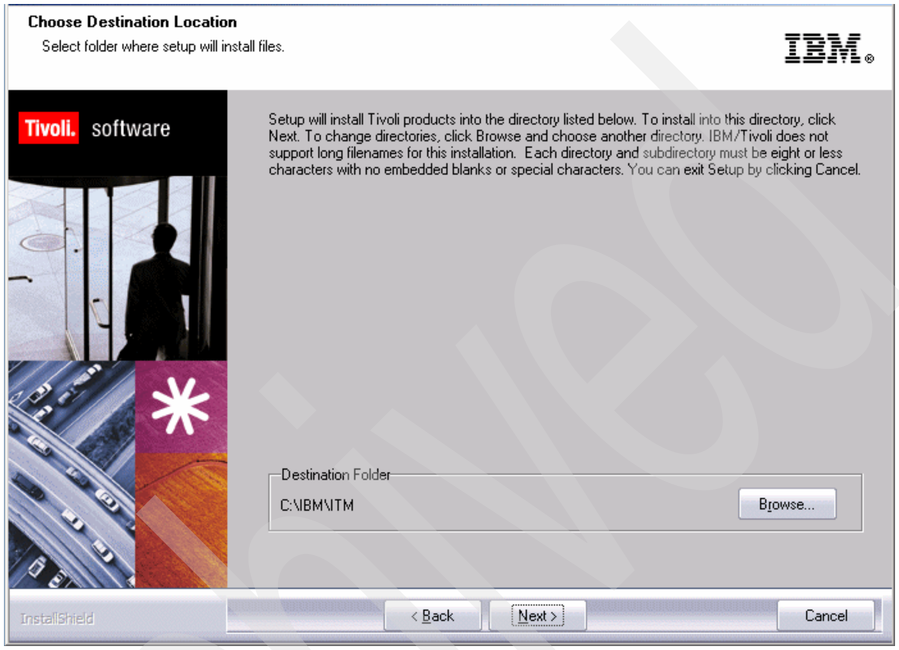
BY DOWNLOADING, INSTALLING, COPYING, ACCESSING, OR USING THE PROGRAM YOU AGREE TO THE TERMS OF THIS AGREEMENT. IF YOU ARE ACCEPTING THESE TERMS ON BEHALF OF ANOTHER PERSON OR A COMPANY OR OTHER LEGAL ENTITY, YOU REPRESENT AND WARRANT THAT YOU HAVE FULL AUTHORITY TO BIND THAT PERSON, COMPANY, OR LEGAL ENTITY TO THESE TERMS. IF YOU DO NOT AGREE TO THESE TERMS,

- DO NOT DOWNLOAD, INSTALL, COPY, ACCESS, OR USE THE PROGRAM; AND

- PROMPTLY RETURN THE PROGRAM TO THE PARTY FROM WHOM YOU ACQUIRED IT. IF YOU DOWNLOADED THE PROGRAM, CONTACT THE PARTY FROM WHOM YOU

Figure 3-3 License agreement windows

4. Choose the directory where you want to install the product. The default directory is C:\IBM\ITM. You are strongly recommended to install IBM Tivoli Monitoring 6.1 in a different drive from one that holds the operating system. Click **Next**. Figure 3-4 shows the windows with the installation directory.


Choose Destination Location

Select folder where setup will install files.

Setup will install Tivoli products into the directory listed below. To install into this directory, click Next. To change directories, click Browse and choose another directory. IBM/Tivoli does not support long filenames for this installation. Each directory and subdirectory must be eight or less characters with no embedded blanks or special characters. You can exit Setup by clicking Cancel.

Destination Folder: C:\IBM\ITM

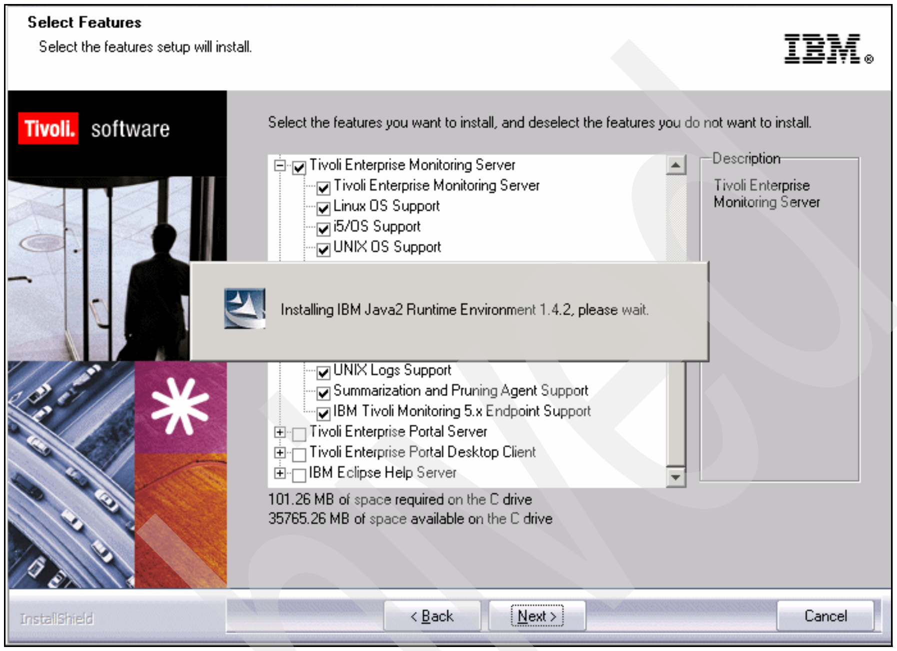
Figure 3-4 Installation windows

5. The next window asks you to type a 32-bit encryption key. You can use the default key.

Notes:

- This encryption key is used to established a secure connection (using SSL protocol) between the Hub TEMS and the other components of the IBM Tivoli Monitoring 6.1 environment as the Remote TEMS connected to the hub. Do not use any of the following characters in your key:
  - =
  - '
  - |
- Ensure that you document the value you use for the key. Use this key during the installation of any components that communicate with this monitoring server.

6. Click **Next**, then **OK** to confirm the encryption key.
7. Select the components that you want to install. Figure 3-5 shows the components we selected for our installation. Click **Next**.



Select Features

Select the features setup will install.

Select the features you want to install, and deselect the features you do not want to install.

- [✓] Tivoli Enterprise Monitoring Server
  - [✓] Tivoli Enterprise Monitoring Server
  - [✓] Linux OS Support
  - [✓] i5/OS Support
  - [✓] UNIX OS Support
  - [✓] UNIX Logs Support
  - [✓] Summarization and Pruning Agent Support
  - [✓] IBM Tivoli Monitoring 5.x Endpoint Support
- [ ] Tivoli Enterprise Portal Server
- [ ] Tivoli Enterprise Portal Desktop Client
- [ ] IBM Eclipse Help Server

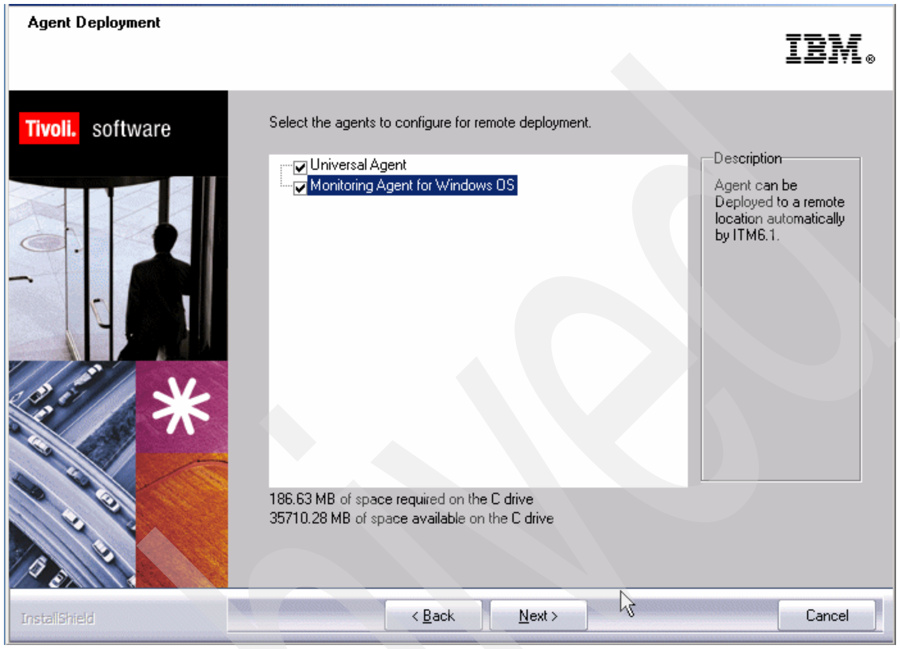
Description: Tivoli Enterprise Monitoring Server

Installing IBM Java2 Runtime Environment 1.4.2, please wait.

101.26 MB of space required on the C drive
35765.26 MB of space available on the C drive

Figure 3-5 List of selected components to be installed

8. If you want to perform remote deployment of agent software, select the agents that you want to deploy (Figure 3-6). This step creates and populates the deployment depot, from which you can deploy agents at a later time.


Agent Deployment

Select the agents to configure for remote deployment.

- [✓] Universal Agent
- [✓] Monitoring Agent for Windows OS

Description

Agent can be Deployed to a remote location automatically by ITM6.1.

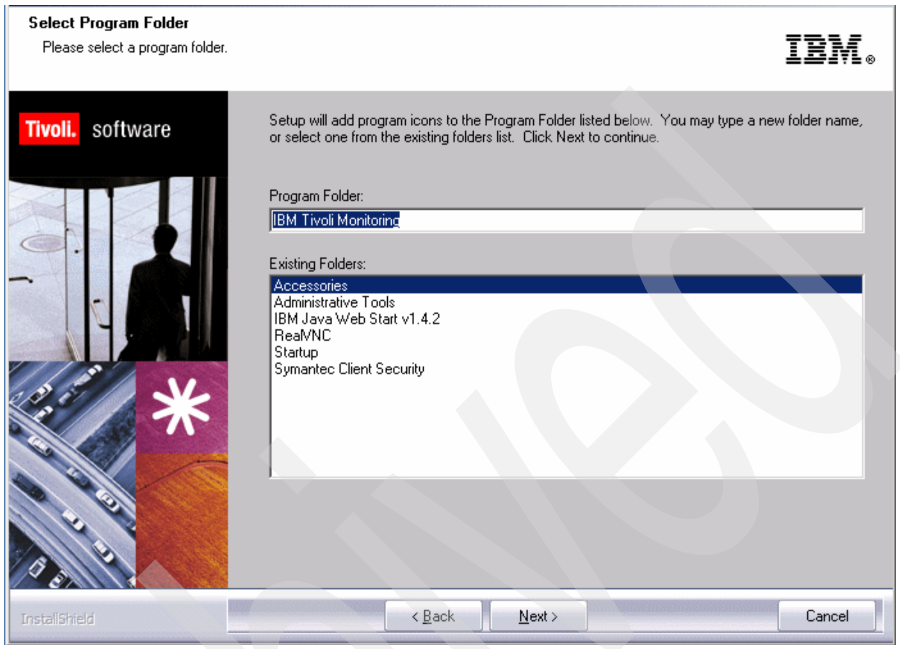
186.63 MB of space required on the C drive
35710.28 MB of space available on the C drive

Figure 3-6 Agent list for remote deployment

9. Click **Next**.

> **Note:** By default, the depot is located in the <itm_installdir>/CMS/depot directory on Windows and <itm_installdir>/tables/<ms_name>/depot directory on Linux and UNIX. If you want to use a different directory, change the DEPOTHOME value in the kbb.env file.

10. Select a program folder as outlined in Figure 3-7 and click **Next**. The default program folder name is IBM Tivoli Monitoring.


Select Program Folder

Please select a program folder.

Setup will add program icons to the Program Folder listed below. You may type a new folder name, or select one from the existing folders list. Click Next to continue.

Program Folder:

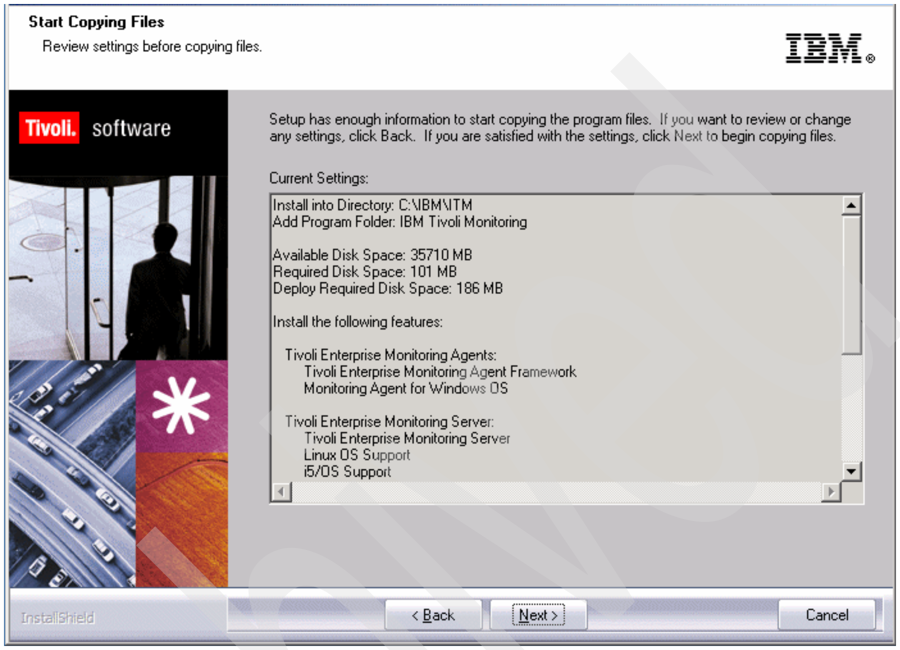
IBM Tivoli Monitoring

Existing Folders:

Accessories
Administrative Tools
IBM Java Web Start v1.4.2
RealVNC
Startup
Symantec Client Security

Figure 3-7 Program Folder for the IBM Tivoli Monitoring 6.1 installation

11. Review the installation summary details. This summary identifies what you are installing and where you have chosen to install. Click **Next** to begin the installation of components (Figure 3-8).


Start Copying Files

Review settings before copying files.

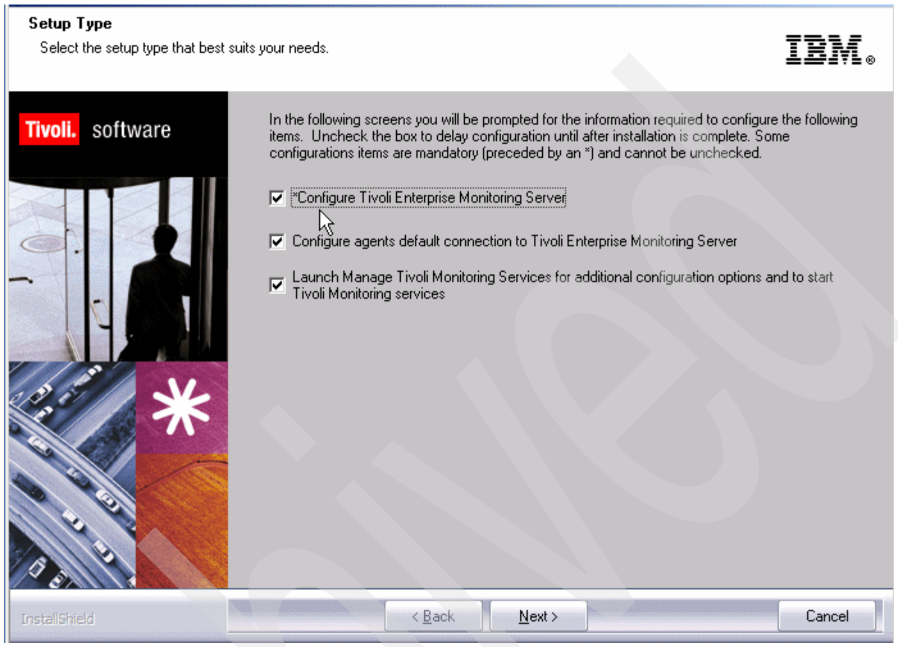
Setup has enough information to start copying the program files. If you want to review or change any settings, click Back. If you are satisfied with the settings, click Next to begin copying files.

Current Settings:

```
Install into Directory: C:\IBM\ITM
Add Program Folder: IBM Tivoli Monitoring

Available Disk Space: 35710 MB
Required Disk Space: 101 MB
Deploy Required Disk Space: 186 MB

Install the following features:

  Tivoli Enterprise Monitoring Agents:
      Tivoli Enterprise Monitoring Agent Framework
      Monitoring Agent for Windows OS

  Tivoli Enterprise Monitoring Server:
      Tivoli Enterprise Monitoring Server
      Linux OS Support
      i5/OS Support
```

Figure 3-8 Installation summary details

12. After the components are installed, a configuration window (Figure 3-9) is displayed with the list of components that can be configured. Select those you want to configure and click **Next**. We selected all three components.



Setup Type

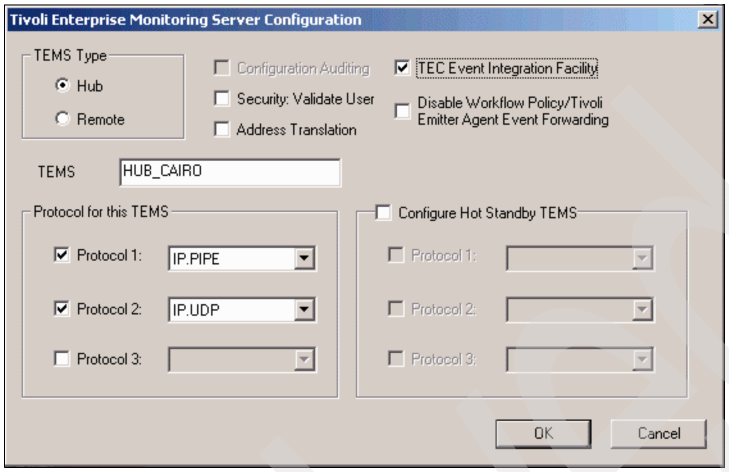
Select the setup type that best suits your needs.

In the following screens you will be prompted for the information required to configure the following items. Uncheck the box to delay configuration until after installation is complete. Some configurations items are mandatory (preceded by an *) and cannot be unchecked.

- [✓] *Configure Tivoli Enterprise Monitoring Server
- [✓] Configure agents default connection to Tivoli Enterprise Monitoring Server
- [✓] Launch Manage Tivoli Monitoring Services for additional configuration options and to start Tivoli Monitoring services

Figure 3-9 List of components that will be configured

13. Figure 3-10 shows the different options of monitoring the type of server that can be selected. Select **Hub**.


Tivoli Enterprise Monitoring Server Configuration

TEMS Type: [✓ Hub] Remote

- [ ] Configuration Auditing
- [ ] Security: Validate User
- [ ] Address Translation
- [x] TEC Event Integration Facility
- [ ] Disable Workflow Policy/Tivoli Emitter Agent Event Forwarding

TEMS: HUB_CAIRO

Protocol for this TEMS:

- [x] Protocol 1: IP.PIPE
- [x] Protocol 2: IP.UDP
- [ ] Protocol 3:

- [ ] Configure Hot Standby TEMS
  - [ ] Protocol 1:
  - [ ] Protocol 2:
  - [ ] Protocol 3:

Figure 3-10 Monitoring server configuration window

14. Verify that the name of this monitoring server is correct in the TEMS field. If it is not, change it. The default name is hub_hostname. We chose HUB_CAIRO as our TEMS name.

15. Identify the communications protocol for the monitoring server. Your choices are: IP.UDP, IP.PIPE, IP.SPIPE, or SNA. You can specify three methods for communication, enabling you to set up backup communication methods. If the method you've identified as Protocol 1 fails, Protocol 2 is used. We selected IP.PIPE as our primary protocol and IP.UDP as secondary one.

Note: IP.PIPE protocol uses TCP, thus, permanent connection is established between the TEMS and the remote servers. This could have an impact on the server performance, because of the number of RPCs that it needs to handle. If using UDP will not cause security breaches in your environment, we recommend that you set up the first protocol as IP.UDP; otherwise use IP.PIPE.

If a firewall is between your TEMS and your agents, you cannot use IP.UDP.

Table 3-5 on page 77 describes the communication protocols that can be used. This information is valid for all components in the IBM Tivoli Monitoring 6.1 environment. We outline only Hub and Remote TEMS in this table.

Table 3-5 Communications protocol descriptions

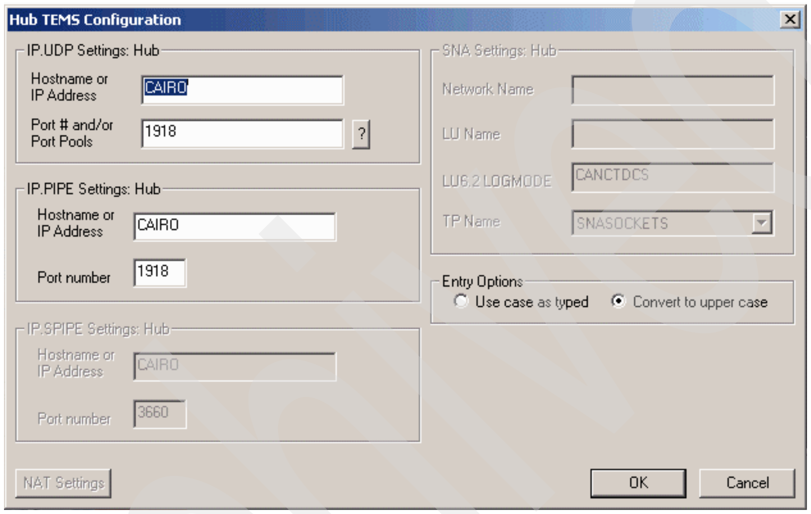
Field	Description
IP.UDP settings: primary Hub TEMSIP.UDP settings: primary Hub TEMS	
Hostname or IP address	The host name or IP address for the hub monitoring server.
Port # and/or Port Pools	The listening port for the hub monitoring server.
IP.PIPE settings: primary Hub TEMS	
Host name or IP Address	The host name or IP address for the hub monitoring server.
Port Number	The listening port for the monitoring server. The default value is 1918.
IP.SPIPE settings: primary Hub TEMS	
Host name or IP Address	The host name or IP address for the hub monitoring server.
Port Number	The listening port for the monitoring server. The default value is 3660.
ESNA settings: remote TEMS	
Local LU Alias	The LU alias
TP Name	The transaction program name for this monitoring server.
SNA settings: primary Hub TEMS	
Network Name	The SNA network identifier for your location.
LU Name	The LU name for the monitoring server. This LU name corresponds to the Local LU Alias in your SNA communications software.
LU 6.2 LOGMODE	The name of the LU6.2 LOGMODE. The default value is .CANCTDCS.
TP Name	The transaction program name for the monitoring server.

16. If you want to forward situation events to IBM Tivoli Enterprise Console, select **TEC Event Integration Facility**.

17. We did not select the Configure Hot Standby TEMS option, because we will set it up when all TEMS are installed and properly configured. Neither did we select Disable Workflow Policy/TivoliEmitter Agent Event Forwarding. We suggest you do the same.

18. Click **OK**.

19. The next window displays the options to configure the hub server name or IP address and communication port the communication protocol will use. Complete the fields as shown in Figure 3-11.



Hub TEMS Configuration

IP.UDP Settings: Hub
- Hostname or IP Address: CAIRO
- Port # and/or Port Pools: 1918

IP.PIPE Settings: Hub
- Hostname or IP Address: CAIRO
- Port number: 1918

IP.SPIPE Settings: Hub
- Hostname or IP Address: CAIRO
- Port number: 3660

SNA Settings: Hub
- Network Name:
- LU Name:
- LU6.2 LOGMODE: CANCTDCS
- TP Name: SNASOCKETS

Entry Options
- Use case as typed [✓ Convert to upper case]

Figure 3-11   Host and communication protocol configuration window

20. If you are certain that you have typed in the values for all of these fields with exactly the correct cases (upper and lower cases), you can select **Use case as typed**. However, because IBM Tivoli Monitoring is case-sensitive, consider selecting **Convert to upper case** to reduce the chance of user error. Click **OK** to continue.

    The next configuration step is to add application support to the monitoring server, such as the workspaces and situations for agents.

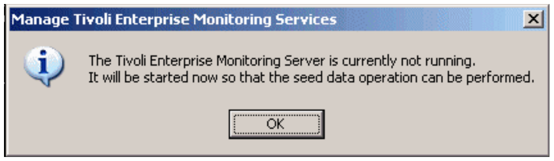
21. After the configuration is complete you are prompted to seed the TEMS. Specify the location of the monitoring server. You have two options:

    a. On this computer

    b. On a different computer

    Chose the first option and click **OK**.

22. Because the monitoring server is not currently running, it will start automatically before the process begins. Click **OK** when you see Figure 3-12.



The Tivoli Enterprise Monitoring Server is currently not running.
It will be started now so that the seed data operation can be performed.

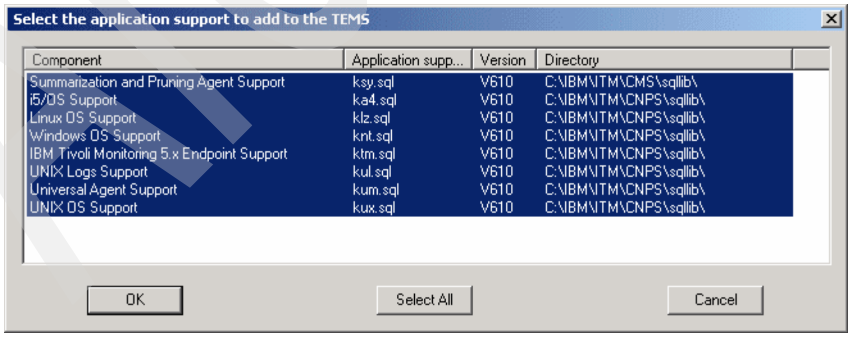
Figure 3-12 Monitoring server start confirmation windows

23. Select the data that you want to add to the monitoring server. By default, all available application support is selected. You are strongly recommended to leave all components selected so that they can be seeded. Click **OK**.

Note: Seeding adds product-specific data from the monitored resources to the monitoring server. For Windows, you can seed the monitoring server both during install and through Manage Tivoli Monitoring Services.

During this process, fields are created in the TEMS database (a flat file/Btrieve database, not the relational database installed for the TEPS) for the agents you have chosen. This enables the TEMS to work with the data from these agents. The same goes for the TEPS, except here of course the necessary tables are created in the relational database of choice.

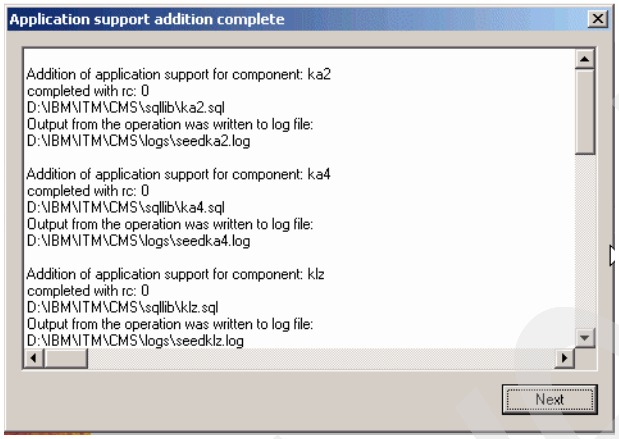
If the seed data is for an agent that reports to a remote monitoring server, complete this process for both the hub and the remote monitoring server. A hub monitoring server should be running before proceeding with a remote monitoring server seed.



Component	Application supp...	Version	Directory
Summarization and Pruning Agent Support	ksy.sql	V610	C:\IBM\ITM\CMS\sqllib\
i5/OS Support	ka4.sql	V610	C:\IBM\ITM\CNPS\sqllib\
Linux OS Support	klz.sql	V610	C:\IBM\ITM\CNPS\sqllib\
Windows OS Support	knt.sql	V610	C:\IBM\ITM\CNPS\sqllib\
IBM Tivoli Monitoring 5.x Endpoint Support	ktm.sql	V610	C:\IBM\ITM\CNPS\sqllib\
UNIX Logs Support	kul.sql	V610	C:\IBM\ITM\CNPS\sqllib\
Universal Agent Support	kum.sql	V610	C:\IBM\ITM\CNPS\sqllib\
UNIX OS Support	kux.sql	V610	C:\IBM\ITM\CNPS\sqllib\

Figure 3-13 Application support to be added to TEMS

24. Verify that each application support added for the components has a return code (rc) equal to 0, as shown in Figure 3-14. Click **Next**.



```
Addition of application support for component: ka2
completed with rc: 0
D:\IBM\ITM\CMS\sqllib\ka2.sql
Output from the operation was written to log file:
D:\IBM\ITM\CMS\logs\seedka2.log

Addition of application support for component: ka4
completed with rc: 0
D:\IBM\ITM\CMS\sqllib\ka4.sql
Output from the operation was written to log file:
D:\IBM\ITM\CMS\logs\seedka4.log

Addition of application support for component: klz
completed with rc: 0
D:\IBM\ITM\CMS\sqllib\klz.sql
Output from the operation was written to log file:
D:\IBM\ITM\CMS\logs\seedklz.log
```

Figure 3-14 Application addition support window

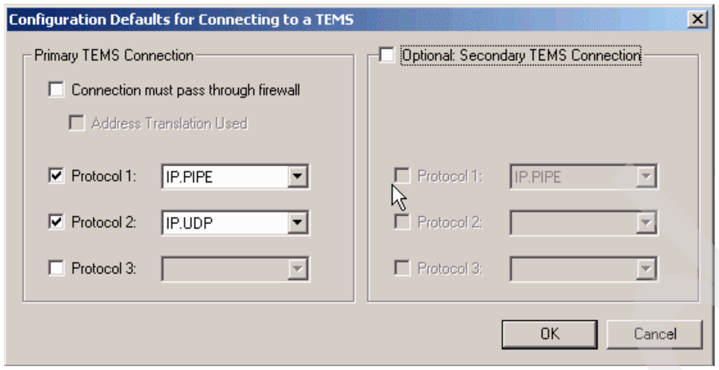
The next configuration step (Figure 3-15 on page 81) configures the default communication between any IBM Tivoli Monitoring component and the hub monitoring server.

25. Specify the default values for IBM Tivoli Monitoring components to use when they communicate with the monitoring server.

    a. If agents must cross a firewall to access the monitoring server, select **Connection must pass through firewall**.

    b. Identify the type of protocol that the agents use to communicate with the hub monitoring server. Your choices are: IP.UDP, IP.PIPE, IP.SPIPE, or SNA as described in Table 3-5 on page 77. You can specify three methods for communication; this enables you to set up backup communication methods. If the method you identified as Protocol 1 fails, Protocol 2 is used. If using UDP will not break your security rules, we suggest using IP.UDP protocol.

    c. Click **OK**.



Configuration Defaults for Connecting to a TEMS

Primary TEMS Connection

- [ ] Connection must pass through firewall
- [ ] Address Translation Used
- [x] Protocol 1: IP.PIPE
- [x] Protocol 2: IP.UDP
- [ ] Protocol 3:

[ ] Optional: Secondary TEMS Connection

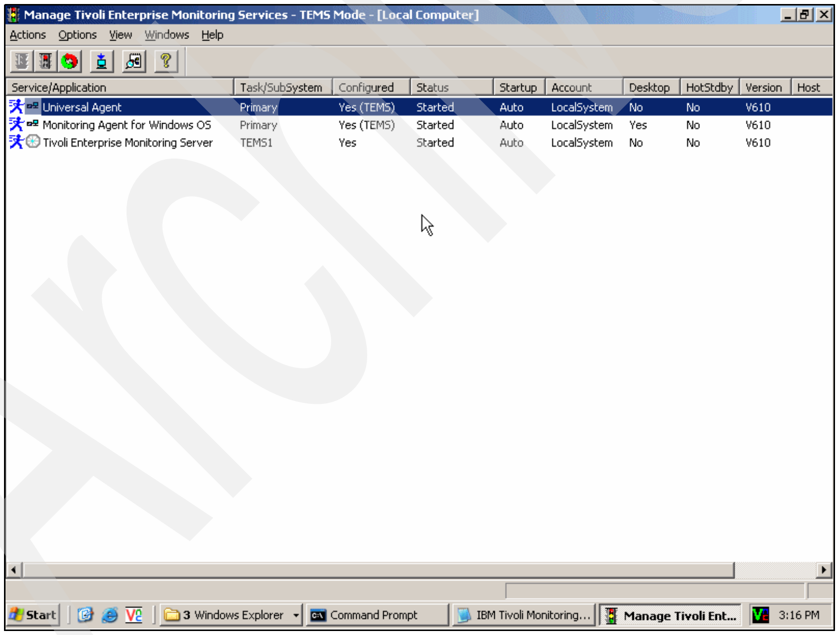
- [ ] Protocol 1: IP.PIPE
- [ ] Protocol 2:
- [ ] Protocol 3:

OK Cancel

Figure 3-15 Communication protocol configuration to a TEMS

26. In the next window, click **Finish** to complete the installation.

    The Manage Tivoli Enterprise Monitoring Services utility opens (Figure 3-16). You can start, stop, and configure IBM Tivoli Monitoring components with this utility.



Manage Tivoli Enterprise Monitoring Services - TEMS Mode - [Local Computer]

Actions Options View Windows Help

Service/Application	Task/SubSystem	Configured	Status	Startup	Account	Desktop	HotStdby	Version	Host
Universal Agent	Primary	Yes (TEMS)	Started	Auto	LocalSystem	No	No	V610	
Monitoring Agent for Windows OS	Primary	Yes (TEMS)	Started	Auto	LocalSystem	Yes	No	V610	
Tivoli Enterprise Monitoring Server	TEMS1	Yes	Started	Auto	LocalSystem	No	No	V610	

Start | 3 Windows Explorer | Command Prompt | IBM Tivoli Monitoring... | Manage Tivoli Ent... | 3:16 PM

Figure 3-16 IBM Tivoli Monitoring 6.1 services window

27. Use this same procedure to install the second Hub TEMS.

3.2.6 Installing a Remote TEMS on a Windows and UNIX server

This section provides detailed information about installing and configuring the remote monitoring server. The following procedures are used in both scenarios.

Installing a Remote TEMS on a Windows server

The installation of a Remote TEMS is similar to the installation of a Hub TEMS. Unless they differ from the Hub TEMS installation, the figures for the Remote TEMS installation will not be shown.

Use the following steps to install the remote monitoring server on a Windows computer:

1. Launch the installation wizard by double-clicking the **setup.exe** file on the installation media.

   Note: If you are running Windows 2003 or Windows XP and have security set to check the software publisher of applications, you might receive an error stating that the setup.exe file is from an unknown publisher. Click **Run** to disregard this error message.

2. Click **Next** on the welcome window.
3. Click **Accept** to accept the license agreement.

   Note: If you do not have a database (DB2 or MS SQL) installed on this computer, a message regarding potentially missing software is displayed. You do not need a database to use this computer as a monitoring server, so you can ignore this message and click **Next**.

4. If you are missing the IBM Java SDK, the installation program installs it automatically during a later step. Click **Next**.
5. Choose the directory where you want to install the product. The default directory is C:\IBM\ITM. Click **Next**.
6. Type a 32-bit encryption key or use the provided default key.

   Note: Ensure that you document the value you use for the key. Use this key during the installation of any components that communicate with this monitoring server.

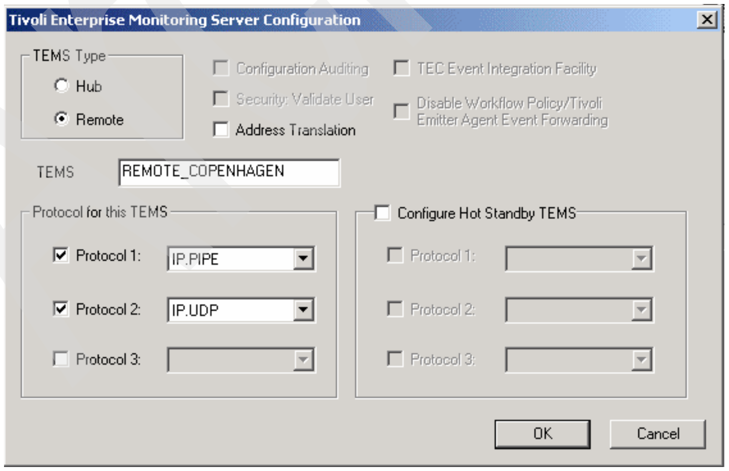
7. Click **Next,** then **OK** to confirm the encryption key.

8. Select the components that you want to install: **Tivoli Enterprise Monitoring Server**.
9. If you want to install any agents on this remote monitoring server, expand Tivoli Enterprise Monitoring Agents and select the agent. Click **Next**.
10. If you want to do remote deployment of agent software from this remote monitoring server, select those agents that you want to deploy. This step creates and populates the deployment depot, from which you can deploy agents at a later time. Click **Next**.

    Note: By default, the depot is located in the <itm_installdir>/CMS/depot directory on Windows and <itm_installdir>/tables/<ms_name>/depot directory on Linux and UNIX. If you want to use a different directory, change the DEPOTHOME value in the kbb.env file.

11. Select a program folder and click **Next**. The default program folder name is IBM Tivoli Monitoring.
12. Review the installation summary details. This summary identifies what you are installing and where you have chosen to install. Click **Next** to start the installation of components.

    After the components are installed, a configuration window opens.
13. Select what you want to configure and click **Next**. The first step configures the monitoring server.
14. Select the type of monitoring server you are configuring: Hub or Remote. For this procedure, select **Remote** as shown in Figure 3-17.


Tivoli Enterprise Monitoring Server Configuration

TEMS Type: ( ) Hub [● Remote]

[ ] Configuration Auditing
[ ] Security: Validate User
[ ] Address Translation
[ ] TEC Event Integration Facility
[ ] Disable Workflow Policy/Tivoli Emitter Agent Event Forwarding

TEMS: REMOTE_COPENHAGEN

Protocol for this TEMS
[✓] Protocol 1: IP.PIPE
[✓] Protocol 2: IP.UDP
[ ] Protocol 3:

[ ] Configure Hot Standby TEMS
[ ] Protocol 1:
[ ] Protocol 2:
[ ] Protocol 3:

Figure 3-17 Remote TEMS configuration window

15. Verify that the name of this monitoring server is correct in the TEMS. If it is not, change it.
16. Identify the communications protocol for the monitoring server. Your choices are: IP.UDP, IP.PIPE, IP.SPIPE, or SNA. You can specify three methods for communication, which enables you to set backup communication methods. If the method you have identified as Protocol 1 fails, Protocol 2 will be used. Click **OK**.
17. Complete the following fields for the communications protocol for the monitoring server. Table 3-5 on page 77 shows descriptions for protocols that can be used.
18. If you are certain that you have typed the values for all of these fields with exactly the correct casing (upper and lower cases), you can select **Use case as typed**. However, because IBM Tivoli Monitoring is case-sensitive, consider selecting **Convert to upper case** to reduce the chance of user error.
19. Click **OK** to continue.

    The next configuration step is to seed the monitoring server.
20. Specify the location of the monitoring server. You have two options:
    a. This computer
    b. On a different computer

    Select **This computer** and click **OK**.
21. Because the monitoring server is not currently running, it is started automatically before the seeding process begins. Click **OK** when you get the the message that tells you this.
22. Select the data that you want to add to the monitoring server. By default, all available product data is selected. Click **OK**.
23. Click **Next** on the message that provides information about the seeding. The next configuration step configures the default communication between any IBM Tivoli Monitoring component and the hub monitoring server.
24. Specify the default values for any IBM Tivoli Monitoring component to use when they communicate with the monitoring server.
    a. If agents must cross a firewall to access the monitoring server, select **Connection must pass through firewall**.
    b. Identify the type of protocol that the agents use to communicate with the hub monitoring server. Your choices are: IP.UDP, IP.PIPE, IP.SPIPE, or SNA. You can specify three methods for communication, which enables you to set backup communication methods. If the method you have identified as Protocol 1 fails, Protocol 2 is used. Click **OK**.

25. Complete the communication protocol fields for the monitoring server. See Table 3-5 on page 77 for definitions of these fields.

26. Click **Finish** to complete the installation.

Installing Remote TEMS on a UNIX/Linux server

The Remote TEMS installation procedure is the same as the one for Hub TEMS. The difference occurs during the configuration. Table 3-6 shows the steps for installing, configuring, and seeding a Remote TEMS.

Table 3-6 Steps for installing a Remote TEMS

Steps	Where to find information
1. Install the Remote TEMS using the same instruction as installing the Hub TEMS.	Installing a Hub TEMS on a UNIX server
2. Configure the remote TEMS.	Configuring Remote TEMS on a UNIX/Linux server
3. Seed the Remote TEMS.	Installing agents support on (seeding) the hub monitoring server

Configuring Remote TEMS on a UNIX/Linux server

Use the following steps to configure the hub monitoring server:

1. At the command line, change to the `opt/IBM/ITM/bin` directory (or the directory where you installed IBM Tivoli Monitoring).
2. Run the following command:

   ```
   ./itmcmd config -S -t tems_name
   ```

   tems_name is the name of your monitoring server (for example, REMOTE_EDINBURG).
3. Type `remote` to indicate that this is a Remote TEMS.
4. Press Enter to accept the default host name for the monitoring server. This should be the host name for your computer. If it is not, type the correct host name and press Enter.
5. Enter the type of protocol to use for communication with the monitoring server. Your choices are: ip, ip.pipe, sna, or ip.spipe. Press Enter to use the default communications protocol (IP.PIPE).
6. If you want to set up a backup protocol, enter that protocol and press Enter. If you do not want to use backup protocol, press Enter without specifying a protocol.

7. Depending on the type of protocol you specified, provide the port number for each communication protocol and press Enter.
8. Press Enter to not specify the name of the KDC_PARTITION.
9. Press Enter when prompted for the path and name of the KDC_PARTITION.
10. If you want to use Configuration Auditing, type `y`; otherwise type `n` and press Enter.
11. Press Enter to accept the default setting for Hot Standby (NO). For best results, wait until after you have fully deployed your environment to configure Hot Standby for your monitoring server. See "Configuring Hot Standby" on page 163 for information about configuring Hot Standby.
12. Press Enter to accept the default for the Optional Primary Network Name (`none`).
13. Press Enter for the default security: `Validate User setting (no)`. If you need to use security validation in your environment, you can enable it after initial configuration is complete.
14. If you will use event synchronization to view situation events, type `y` and press Enter to enable TEC Event Integration. Complete the following additional steps:
    a. Type the name of the IBM Tivoli Enterprise Console event server and press Enter.
    b. Type the port number for the event server and press Enter.
15. Press Enter to not disable the Workflow Policy/Tivoli Emitter Agent.
16. Type `S` to save the default SOAP configuration and exit the configuration.

Notes:

- You can configure any SOAP information at a later time. The procedure is described in "Installing event synchronization on your event server" on page 149.
- A configuration file is generated in the install_dir/config directory with the format host_name_ms_tems_name.config (for example, edinburg_ms_REMOTE_EDINBURG.config).

3.2.7 Tivoli Enterprise Portal Server - TEPS

This section describes the steps for installing and configuring a TEPS in a Windows server using DB2 8.2 as RDBMS.

Note: Our choice for the Windows server was driven by a limitation of the IBM Tivoli Monitoring 6.1 beta version we were using. In the beta version of the code that we used for this project, it was not possible to connect to a Data Warehouse for long-term historical data views using a Linux portal server. This limitation is expected to go away in the general availability version. Refer the IBM Tivoli Monitoring 6.1 general availability documentation or contact IBM support to verify whether this issue has been resolved, if you are considering changing your platform because of that limitation.

Preinstallation steps

The Tivoli Enterprise Portal Server function requires a database to store information. Refer to *IBM Tivoli Monitoring Installation and Setup Guide*, GC32-9407, to install and set up your RDBMS.

Install the RDBMS

Prior to installing the TEPS, you have to install and set up the DB2 database in your environment.

ODBC connection for the TEPS

TEPS access the created database using ODBC connection. ODBC TEPS2 will be created during TEPS installation, so it is not necessary to create it manually.

Create a user on the DB2 server

Create a DB2 user in the DB2 server. You can use any name you want but the user must belong to Administrators group. We created a user named ITMUser.

Note: This user will be used by TEPS to access the Data Warehouse.

Installing the portal server on a Window server

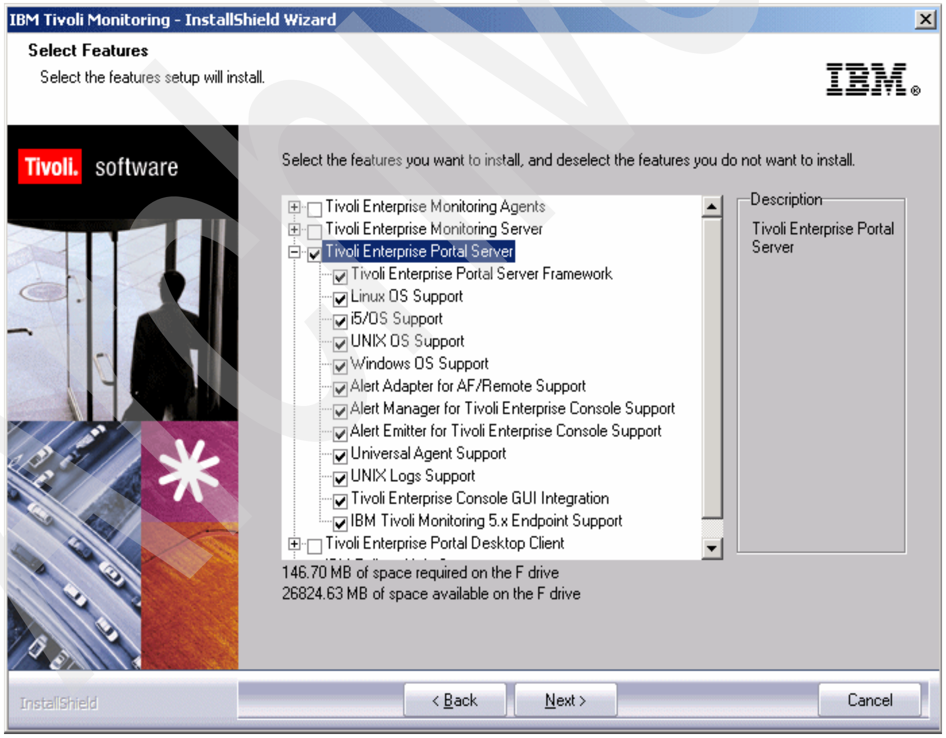
Use the following steps to install the Tivoli Enterprise Portal Server on a Windows computer:

1. Launch the installation wizard by double-clicking the **setup.exe** file in the WINDOWS subdirectory of the installation media.
2. Click **Next** on the Welcome window to start the installation.
3. Read and accept the software license agreement by clicking **Accept**.

4. If you do not have a database (DB2 or MS SQL) or the IBM Java SDK installed on this computer, a message regarding potentially missing required software is displayed. If you are missing a database, stop the installation, install the required database, and begin the installation again. If you are missing the IBM Java SDK, the installation program installs it automatically during a later step. Click **Next**.

   Note: If your computer has all required software, you will not see this step.

5. Specify the directory where you want to install the portal software and accompanying files. The default location is C:\IBM\ITM. Click **Next**.
6. Type an encryption key to use. This key should be the same as what was used during the installation of the monitoring server to which this portal server will connect. Click **Next** and then **OK** to confirm the encryption key.
7. Select Tivoli Enterprise Portal Server from the list of components to install as shown in Figure 3-18.


IBM Tivoli Monitoring - InstallShield Wizard

Select Features

Select the features setup will install.

Select the features you want to install, and deselect the features you do not want to install.

- ☐ Tivoli Enterprise Monitoring Agents
- ☐ Tivoli Enterprise Monitoring Server
- ☑ Tivoli Enterprise Portal Server
  - ☑ Tivoli Enterprise Portal Server Framework
  - ☑ Linux OS Support
  - ☑ i5/OS Support
  - ☑ UNIX OS Support
  - ☑ Windows OS Support
  - ☑ Alert Adapter for AF/Remote Support
  - ☑ Alert Manager for Tivoli Enterprise Console Support
  - ☑ Alert Emitter for Tivoli Enterprise Console Support
  - ☑ Universal Agent Support
  - ☑ UNIX Logs Support
  - ☑ Tivoli Enterprise Console GUI Integration
  - ☑ IBM Tivoli Monitoring 5.x Endpoint Support
- ☐ Tivoli Enterprise Portal Desktop Client

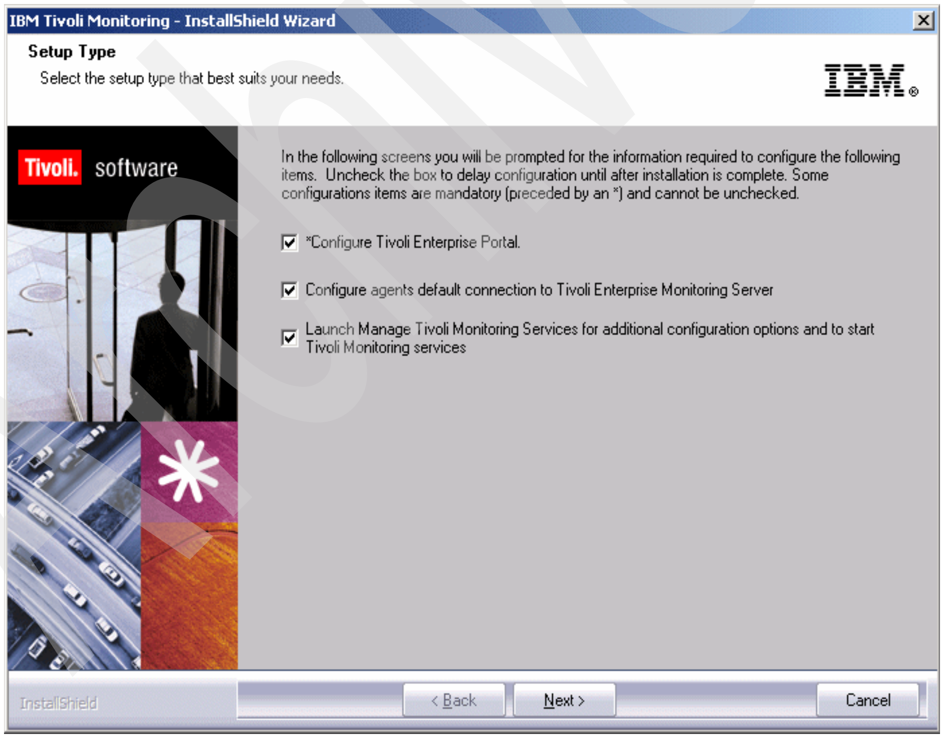
Description: Tivoli Enterprise Portal Server

146.70 MB of space required on the F drive
26824.63 MB of space available on the F drive

Figure 3-18 IBM Tivoli Monitoring 6.1 Components List

Note: You might notice that it will begin installing the required JRE on the machine as soon as you select the TEPS for installation. This happens if you do not have the required JRE installed in your machine. The JRE is bundled with the installation media and you do not have to do anything but wait until the installation is finished.

8. If you want to view events from the IBM Tivoli Enterprise Console event server through the Tivoli Enterprise Portal, expand the **Tivoli Enterprise Portal Server** selection and ensure that **Tivoli Enterprise Console GUI Integration** is selected. Click **Next**.
9. Do not select any agents on the Agent Deploy window. Click **Next**.
10. Type a name for the program folder. The default is IBM Tivoli Monitoring. Click **Next**.
11. Click **Next** to start the installation. After installation is complete, a configuration window (Table 3-19) is displayed.


IBM Tivoli Monitoring - InstallShield Wizard
Setup Type
Select the setup type that best suits your needs.
IBM.
Tivoli. software
InstallShield
< Back
Next >
Cancel

In the following screens you will be prompted for the information required to configure the following items. Uncheck the box to delay configuration until after installation is complete. Some configurations items are mandatory (preceded by an *) and cannot be unchecked.

☑ *Configure Tivoli Enterprise Portal.

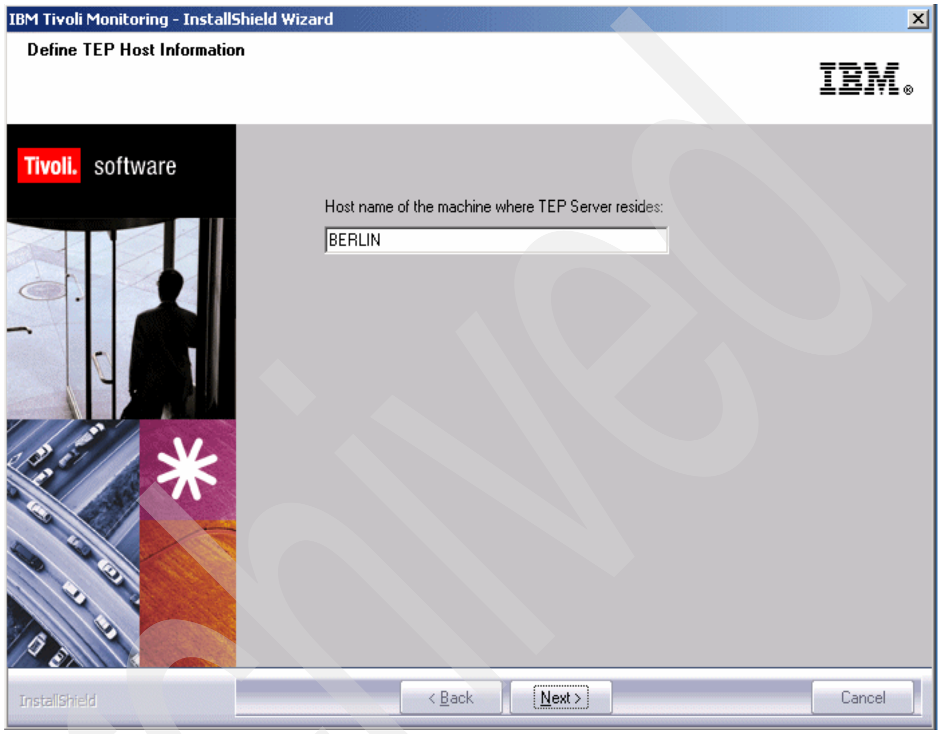
☑ Configure agents default connection to Tivoli Enterprise Monitoring Server

☑ Launch Manage Tivoli Monitoring Services for additional configuration options and to start Tivoli Monitoring services

Figure 3-19 TEPS configuration option window

12. Click **Next** to begin configuring the portal server and the connection to the monitoring server, and to open Manage Tivoli Monitoring Services.

13. In the next window (Table 3-20) type the host name of the computer where you are installing the portal server and click **Next**.


IBM Tivoli Monitoring - InstallShield Wizard
Define TEP Host Information
IBM
Tivoli. software
InstallShield
< Back
Next >
Cancel

Host name of the machine where TEP Server resides:

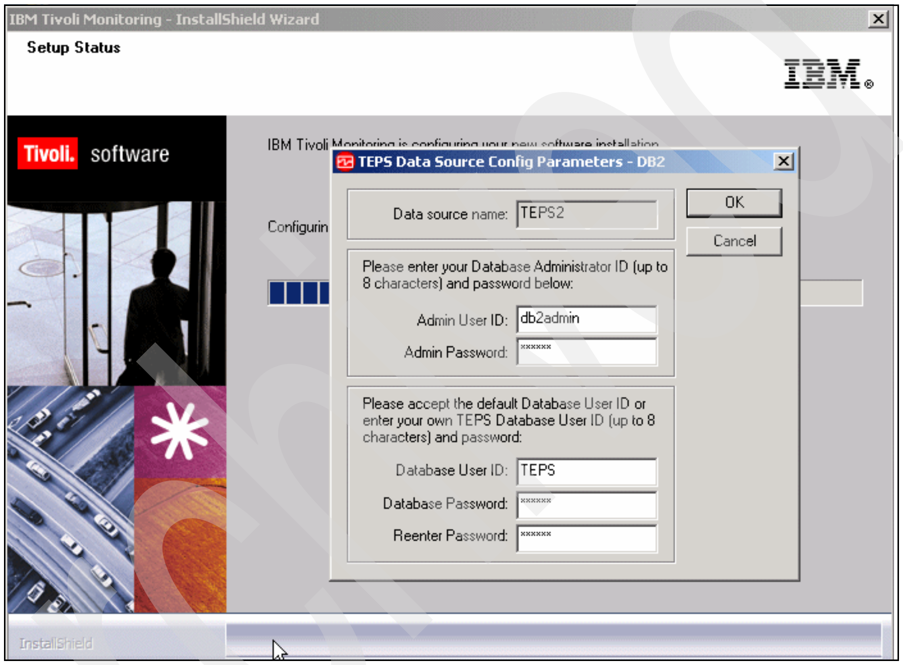
BERLIN

Figure 3-20 Hostname where TEPS will be installed

14. Configure the portal server's connection to the data source (such as your DB2 database). Type the password for the database administrator in the Admin Password field, as shown in Figure 3-21.

15. Type a database user ID and password for your db2 Administrator account and click **OK**.

> **Note:** DB2 Administrator account was created during the DB2 installation.


IBM Tivoli Monitoring - InstallShield Wizard
Setup Status
IBM
Tivoli. software
IBM Tivoli Monitoring is configuring your new software installation.
Configurin
InstallShield

TEPS Data Source Config Parameters - DB2

Data source name: TEPS2

Please enter your Database Administrator ID (up to 8 characters) and password below:

Admin User ID: db2admin

Admin Password: ******

Please accept the default Database User ID or enter your own TEPS Database User ID (up to 8 characters) and password:

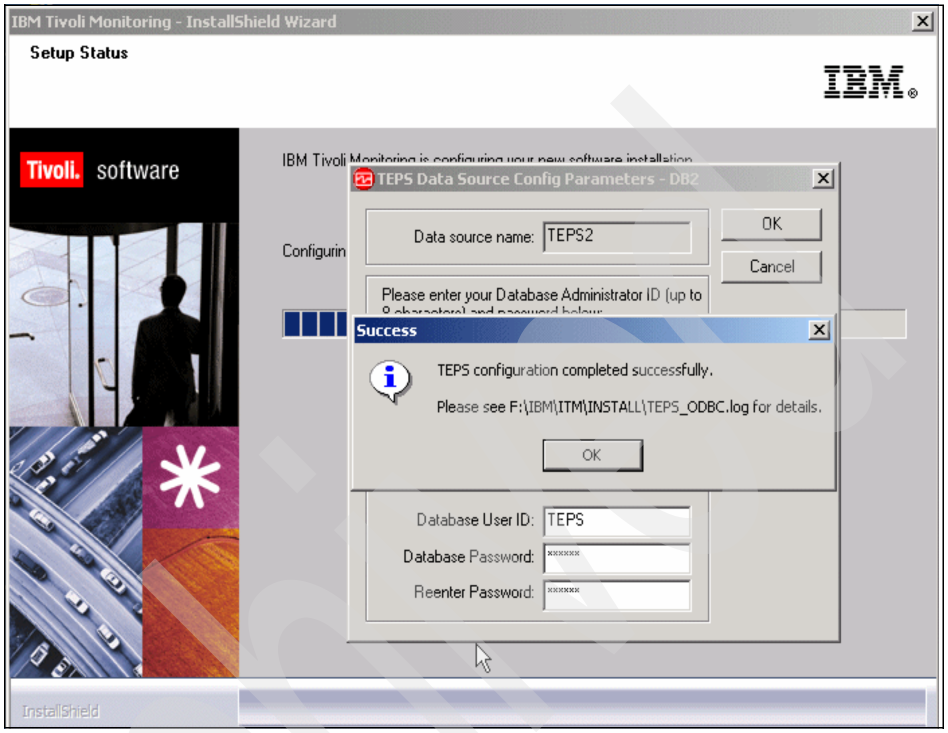
Database User ID: TEPS

Database Password: ******

Reenter Password: ******

Figure 3-21 TEPS database configuration

16. Click **OK** on the message that tells you that the portal server configuration was successful (Figure 3-22).


IBM Tivoli Monitoring - InstallShield Wizard

Setup Status

IBM Tivoli Monitoring is configuring your new software installation.

TEPS Data Source Config Parameters - DB2

Data source name: TEPS2

Please enter your Database Administrator ID (up to 8 characters) and password below:

Success

TEPS configuration completed successfully.

Please see F:\IBM\ITM\INSTALL\TEPS_ODBC.log for details.

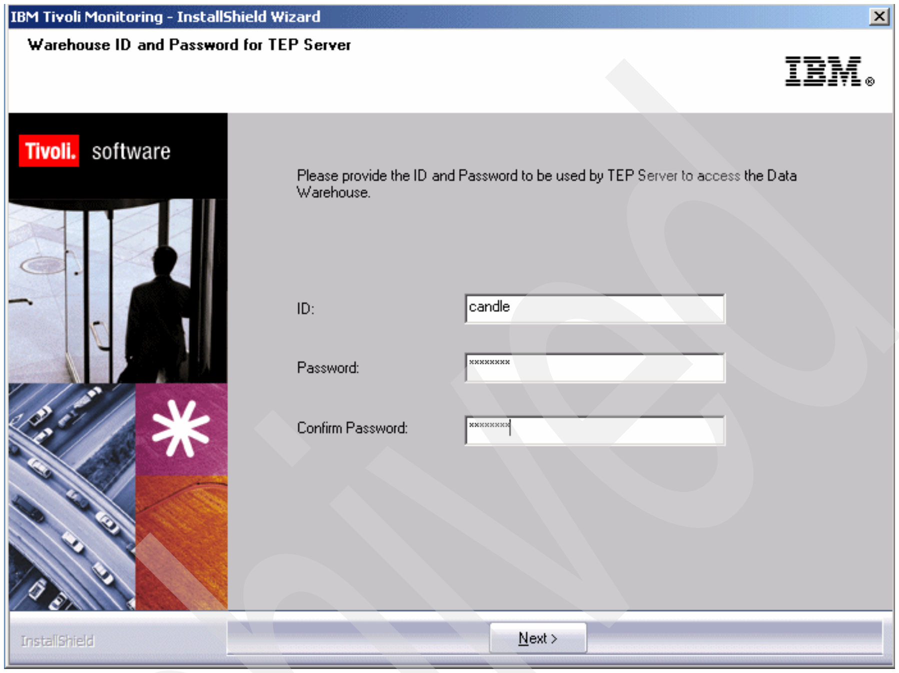
Database User ID: TEPS

Database Password: ******

Reenter Password: ******

Figure 3-22 TEPS configuration completion window

17. Next it asks about the user credentials to access the Data Warehouse database. Type a previously created user ID (such as `candle`) and a password. Click **Next (**Figure 3-23**)**.


IBM Tivoli Monitoring - InstallShield Wizard

Warehouse ID and Password for TEP Server

Please provide the ID and Password to be used by TEP Server to access the Data Warehouse.

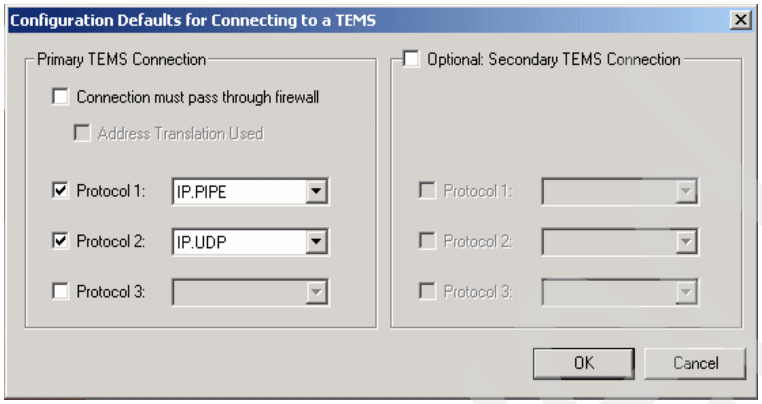
ID: candle

Password: *********

Confirm Password: *********

Figure 3-23 TEPS user configuration

18. Select the communication protocols as shown in Figure 3-24 and click **OK** (Figure 3-24). This defines the values for the connection between the portal server and the hub monitoring server.

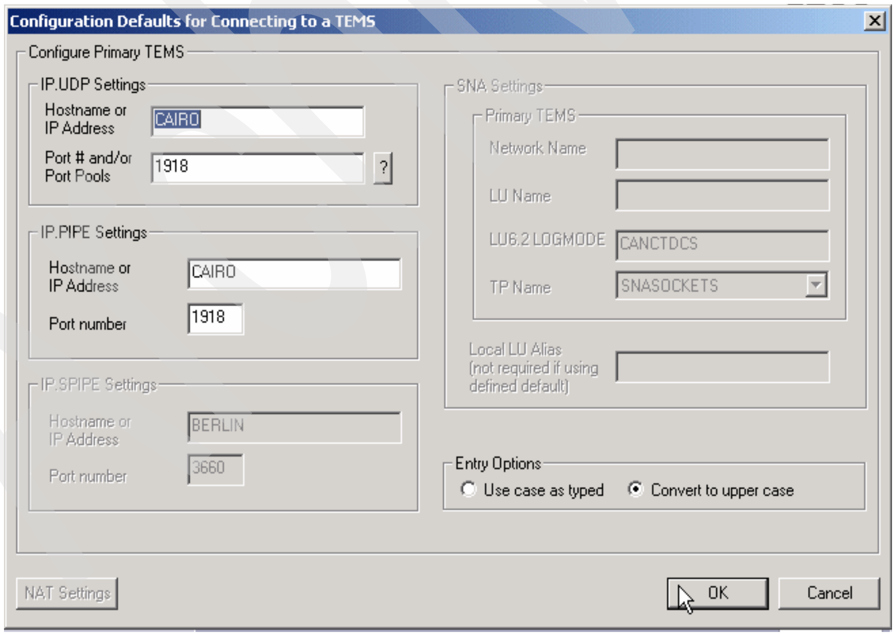

Configuration Defaults for Connecting to a TEMS

Primary TEMS Connection
- [ ] Connection must pass through firewall
- [ ] Address Translation Used
- [✓] Protocol 1: IP.PIPE
- [✓] Protocol 2: IP.UDP
- [ ] Protocol 3:

[ ] Optional: Secondary TEMS Connection
- [ ] Protocol 1:
- [ ] Protocol 2:
- [ ] Protocol 3:

Figure 3-24 Communication protocol window configuration

19. Type the host name or IP address and the port number for the hub monitoring server as shown in Figure 3-25. Click **OK** when finished.


Configuration Defaults for Connecting to a TEMS

Configure Primary TEMS

IP.UDP Settings
- Hostname or IP Address: CAIRO
- Port # and/or Port Pools: 1918

IP.PIPE Settings
- Hostname or IP Address: CAIRO
- Port number: 1918

IP.SPIPE Settings
- Hostname or IP Address: BERLIN
- Port number: 3660

SNA Settings

Primary TEMS
- Network Name:
- LU Name:
- LU6.2 LOGMODE: CANCTDCS
- TP Name: SNASOCKETS

Local LU Alias (not required if using defined default):

Entry Options: Use case as typed [✓ Convert to upper case]

Figure 3-25 Configuration for connection to the TEMS

20. Click **Finish** to close the installation wizard. After the installation completes, a README about Tivoli Enterprise Portal configuration is displayed. Read it and close the window.

Now the Hub TEMS, Remote TEMS, and the TEPS are installed and configured. We will proceed with the installation of the TEMAs and the TEP, Warehouse Proxy agent, and event synchronization.

3.2.8 Tivoli Enterprise Monitoring Agent

In this section we cover the installation of Tivoli Enterprise Monitoring Agents (TEMA).

Deploying TEMA on a Linux server (using local images)

This section shows step by step how we deploy a TEMA on a Linux server using local downloaded IBM Tivoli Monitoring 6.1 images. Our server name is oslo and the user is itmuser.

Note: Whenever we did not enter or select an option, the Enter key was pressed to accept the default.

1. From the directory where the images are uncompressed, execute the following procedure:

   ```
   itmuser@oslo:/home/itmuser> ./install.sh
   ```

   Example 3-1 shows the output of the command with our selections in bold.

Example 3-1 Output of ./install.sh

```
Enter the name of the IBM Tivoli Monitoring directory
[ default = /opt/IBM/ITM ]:
CANDLEHOME directory "/opt/IBM/ITM" already exists.
OK to use it [ y or n; "y" is default ]?  y

Before installing IBM Tivoli Monitoring agents, you must
install at least one IBM Tivoli Enterprise Monitoring Server.  You
will need the host name or IP address and port number for the
monitoring server to configure any agents.
```

Notes:

- If /opt/IBM/ITM does not exist, you will receive the following message:

  ```
  "/opt/IBM/ITM" does not exist
  try to create it [y or n; "y" is default]?
  ```

- You will notice small differences on the message between the following menu with the one you will be running because of the version differences. But the functions remains the same.

Example 3-2 Output of ./install.sh

```
install.sh       :  searching for product families; please wait.
Select one of the following:
1) Install products via command line.
2) Install products to depot via command line.
3) Exit install.
Please enter a valid number:  1
install.sh       :  OK to install.
install.sh       :  removing old JRE.
install.sh       :  old JRE has been removed.
install.sh       :  unloading JRE package(s).
install.sh       :  calculating available disk space.
install.sh       :  "33631276" kilobytes available.
install.sh       :  running li6243 jre.
Software Licensing Agreement
1. Czech
2. English
3. French
4. German
5. Italian
6. Polish
7. Portuguese
8. Spanish
9. Turkish
Please enter the number that corresponds to the language you prefer.
2
Software Licensing Agreement
Press Enter to display the license agreement on your
screen. Please read the agreement carefully before
installing the Program. After reading the agreement, you
will be given the opportunity to accept it or decline it.
If you choose to decline the agreement, installation will
not be completed and you will not be able to use the
Program.
```

2. Press the Enter key.

Example 3-3 Output of ./install.sh

```
International License Agreement for Early Release of Programs

Part 1 - General Terms

BY DOWNLOADING, INSTALLING, COPYING, ACCESSING, OR USING
THE PROGRAM YOU AGREE TO THE TERMS OF THIS AGREEMENT. IF
YOU ARE ACCEPTING THESE TERMS ON BEHALF OF ANOTHER PERSON
OR A COMPANY OR OTHER LEGAL ENTITY, YOU REPRESENT AND
WARRANT THAT YOU HAVE FULL AUTHORITY TO BIND THAT PERSON,
COMPANY, OR LEGAL ENTITY TO THESE TERMS. IF YOU DO NOT
AGREE TO THESE TERMS,

- DO NOT DOWNLOAD, INSTALL, COPY, ACCESS, OR USE THE
PROGRAM; AND
1
runGSkit        :  Preparing to install the Global Security Kit.
runGSkit warning:  the 'root' ID or password is required for this phase,
continuing ..
 Will enable automatic agent initiation after reboot.
Please enter root password or press Enter twice to skip.
Password:
Enter the root password

Preparing packages for installation...
gsk7bas-7.0-3.9
runGSkit        :  creating security files.
runGSkit        :  create keyfile.
runGSkit        :  create certificate.
runGSkit        :  setting encryption key.

Enter a 32-character encryption key, or just press Enter to use the default
        Default = IBMTivoliMonitoringEncryptionKey

....+....1....+....2....+....3..

Press Enter to continue viewing the license agreement, or,
Enter "1" to accept the agreement, "2" to decline it or
"99" to go back to the previous screen.
1
GSkit encryption key has been set.
Key File directory: /opt/IBM/ITM/keyfiles

 Product packages are available in /home/itmuser/unix
   Product packages are available for the following operating systems and
   component support categories:
```

```
 1) Linux Intel R2.4 (32 bit)
 2) Linux Intel R2.4 (64 bit)
 3) Linux Intel R2.4 GCC 2.9.5 (32 bit)
 4) Linux Intel R2.4 GCC 2.9.5 (64 bit)
 5) Linux Intel R2.6 (32 bit)
 6) Linux Intel R2.6 (64 bit)
 7) Linux Intel R2.6 GCC 2.9.5 (32 bit)
 8) Linux Intel R2.6 GCC 2.9.5 (64 bit)
 9) Tivoli Enterprise Portal Browser Client support
10) Tivoli Enterprise Portal Desktop Client support
11) Tivoli Enterprise Portal Server support

Type the number for the OS or component support category you want, or type
"q" to quit selection
[ number "5" or "Linux Intel R2.6 (32 bit)" is default ]:
5

Is the operating system or component support correct [ y or n; "y" is
default ]? y

The following products are available for installation:

 1) IBM Eclipse Help Server V610R104
 2) Monitoring Agent for Linux OS V610R115
 3) Monitoring Agent for Unix Logs V610R121
 4)Summarization and Pruning agent V610R141
 5) Tivoli Enterprise Monitoring Server V610R215
 6) Tivoli Enterprise Portal Desktop Client V610R172
 7) Tivoli Enterprise Portal Server V610R172
 8) Tivoli Enterprise Services User Interface V610R194
 9) Universal Agent V610R229
10) all of the above

Type the numbers for the products you want to install, or type "q" to
quit selection.
If you enter more than one number, separate the numbers by a comma or
a space.

Type your selections here:  2

The following products will be installed:

  Monitoring Agent for Linux OS V610R115

Are your selections correct [ y or n; "y" is default ]? y

 ... installing "Monitoring Agent for Linux OS V610R115 for Linux Intel
R2.6 (32 bit)"; please wait.
```

```
    => installed "Monitoring Agent for Linux OS V610R115 for Linux Intel R2.6
   (32 bit)."
    ... Initializing database for Monitoring Agent for Linux OS V610R115 for
   Linux Intel R2.6 (32 bit).
    ... Monitoring Agent for Linux OS V610R115 for Linux Intel R2.6 (32 bit)
   initialized.

   Do you want to install additional products or product support packages [ y
   or n; "n" is default ]?n

   ... postprocessing; please wait.

    ... finished postprocessing.

    Installation step complete.

You must install TEMS support for the agent products.This is done by starting
and seeding the TEMS for the supported agents.

You may now reconfigure any installed IBM Tivoli Monitoring product via the
"/opt/IBM/ITM/bin/itmcmd config" command.
```

Post TEMA installation procedure

1. From $CANDLEHOME/bin (/opt/Tivoli/IBM is the default $CANDLEHOME directory) execute the following line:

   ```
   itmuser@oslo:/opt/IBM/ITM/bin> ./itmcmd config -A lz
   ```

   Example 3-4 shows the output of the command.

Example 3-4 Post TEMA installation procedure

```
lz for linux, ux for unix
CandleConfig        :  installer level  400 / 100.
CandleConfig        :  running li6243 jre.
Agent configuration started...

Will this agent connect to a TEMS? [YES or NO] (Default is: YES):YES
TEMS Host Name (Default is: oslo): edinburg
```

Note: `edinburg` is the Remote TEMS to which the agent will connect.

Example 3-5 Post TEMA installation procedure

```
Will the agent connect through a firewall? [YES or NO] (Default is: NO):NO

Network Protocol [ip, sna, ip.pipe or ip.spipe] (Default is: ip.pipe):ip.pipe
```

```
     Now choose the next protocol from one of these:
     - ip
     - sna
     - ip.pipe
     - none
Network Protocol 2 (Default is: none): ip

     Now choose the next protocol from one of these:
     - ip
     - sna
     - none
Network Protocol 3 (Default is: none):none
IP Port Number (Default is: 1918):1918
IP.PIPE Port Number (Default is: 1918):1918
Enter name of KDC_PARTITION (Default is: null):null
IP.SPIPE Port Number (Default is: 3660)

Configure connection for a secondary TEMS? [YES or NO] (Default is: NO): YES
Secondary TEMS HostName (Default is: none): copenhagen

Will the agent connect through a firewall? [YES or NO] (Default is: NO):NO

Secondary TEMS protocol [ip, sna, or ip.pipe] (Default is: ip): ip.pipe

     Now choose the next protocol from one of these:
     - ip
     - sna
     - ip.pipe
     - none
Secondary TEMS Protocol 2 (Default is: none): ip

     Now choose the next protocol from one of these:
     - ip
     - sna
     - none
Secondary TEMS Protocol 3 (Default is: none):none
Secondary TEMS IP Port Number (Default is: 1918):1918
Secondary TEMS IP.PIPE Port Number (Default is: 1918):1918
Enter Optional Primary Network Name or "none" (Default is: none):none
Agent configuration completed...
```

Note: When installing on AIX, it asks:

```
Are you installing this product into a clustered environment (Default is:
NO)
```

2. When you are done, you can start the agent using this on the command line:

```
./itmcmd agent start lz
```

3. If you do not know the agent code, execute the **./cinfo** command from $CANGDELHOME/bin directory. You should see something like Example 3-6.

Example 3-6 Output ./cinfo command

```
[root@ankara bin]# ./cinfo

*********** Fri Sep 23 15:31:15 EDT 2005 ******************
User      : root         Group: root bin daemon sys adm disk wheel
Host name : ankara.itsc.austin.ibm.com   Installer Lvl: 400 / 100
CandleHome: /opt/IBM/ITM
***********************************************************
-- CINFO Menu --
 1) Show products installed in this CandleHome
 2) Show which products are currently running
 3) Show configuration settings
 4) Show installed CD release versions
 X) Exit CINFO
```

4. Select option 1 for the type of result shown in Example 3-7.

Example 3-7 Post TEMA installation procedure

```
*********** Fri Sep 23 15:33:37 EDT 2005 ******************
User      : root         Group: root bin daemon sys adm disk wheel
Host name : ankara.itsc.austin.ibm.com   Installer Lvl: 400 / 100
CandleHome: /opt/IBM/ITM
***********************************************************
...Product inventory
ax      IBM Tivoli Monitoring Shared Libraries
         li6243  Version: 610 Rel: 221
jr      Tivoli Enterprise-supplied JRE
         li6243  Version: 400 Rel: 100
lz      Monitoring Agent for Linux OS
         li6263  Version: 610 Rel: 115
uf      Universal Agent Framework
         li6243  Version: 610 Rel: 100
ui      Tivoli Enterprise Services User Interface
         li6243  Version: 610 Rel: 194
um      Universal Agent
         li6243  Version: 610 Rel: 229
-- CINFO Menu --
 1) Show products installed in this CandleHome
 2) Show which products are currently running
 3) Show configuration settings
 4) Show installed CD release versions
 X) Exit CINFO
```

Note: You can get the same result executing the following command:

```
./cinfo -i
```

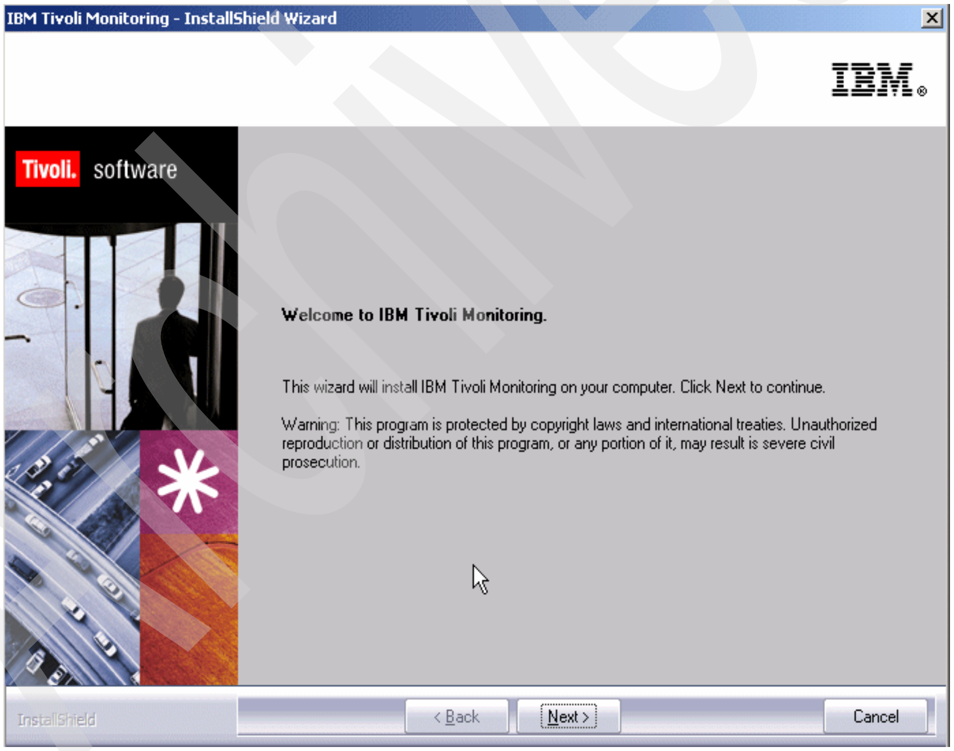
5. If you have installed the Universal Agent as well, you can start executing the following command:

```
./itmcmd config -A um
```

Installing TEMA on a Windows server

Use the following steps to install a monitoring agent:

1. Launch the installation wizard by double-clicking the **setup.exe** file on the installation media.
2. Click **Next** on the welcome window (Figure 3-26).



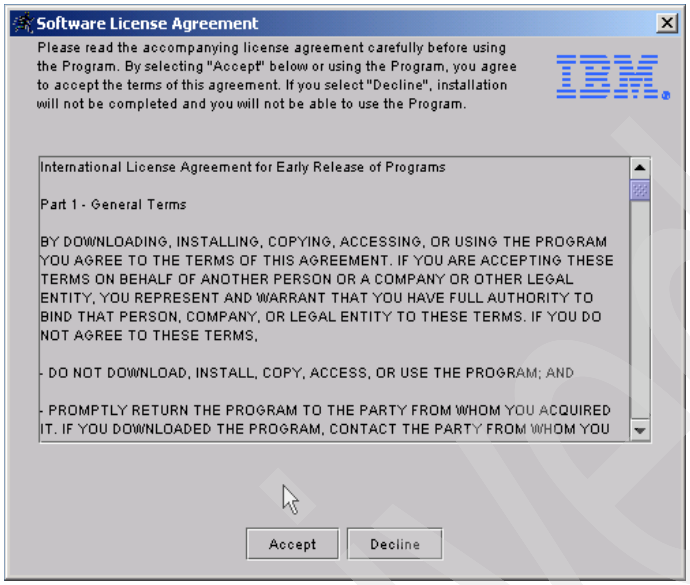
Welcome to IBM Tivoli Monitoring.

This wizard will install IBM Tivoli Monitoring on your computer. Click Next to continue.

Warning: This program is protected by copyright laws and international treaties. Unauthorized reproduction or distribution of this program, or any portion of it, may result is severe civil prosecution.

Figure 3-26 IBM Tivoli Monitoring 6.1 welcome installation window

3. Click **Accept** to accept the license agreement (Figure 3-27).


Software License Agreement

Please read the accompanying license agreement carefully before using the Program. By selecting "Accept" below or using the Program, you agree to accept the terms of this agreement. If you select "Decline", installation will not be completed and you will not be able to use the Program.

International License Agreement for Early Release of Programs

Part 1 - General Terms

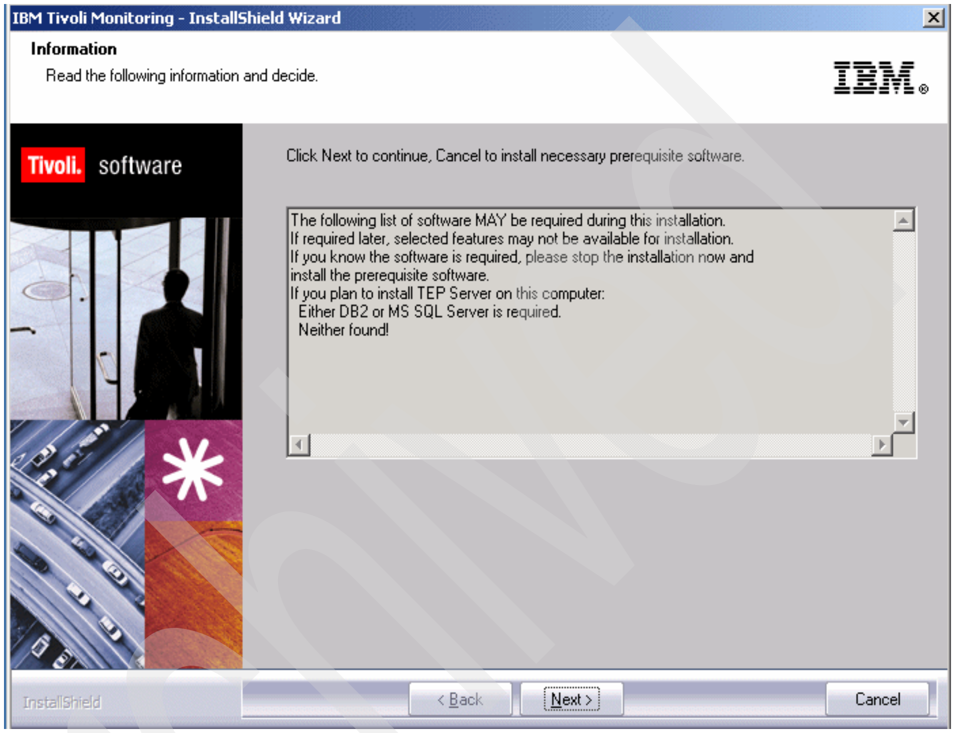
BY DOWNLOADING, INSTALLING, COPYING, ACCESSING, OR USING THE PROGRAM YOU AGREE TO THE TERMS OF THIS AGREEMENT. IF YOU ARE ACCEPTING THESE TERMS ON BEHALF OF ANOTHER PERSON OR A COMPANY OR OTHER LEGAL ENTITY, YOU REPRESENT AND WARRANT THAT YOU HAVE FULL AUTHORITY TO BIND THAT PERSON, COMPANY, OR LEGAL ENTITY TO THESE TERMS. IF YOU DO NOT AGREE TO THESE TERMS,

- DO NOT DOWNLOAD, INSTALL, COPY, ACCESS, OR USE THE PROGRAM; AND

- PROMPTLY RETURN THE PROGRAM TO THE PARTY FROM WHOM YOU ACQUIRED IT. IF YOU DOWNLOADED THE PROGRAM, CONTACT THE PARTY FROM WHOM YOU

Figure 3-27 IBM Tivoli Monitoring 6.1 license agreements

4. If a database (DB2 or MS SQL) is not installed on this computer, a message regarding potentially missing software is displayed as shown in Figure 3-28. You do not need a database to install a management agent on this computer, so you can ignore this message and click **Next**.

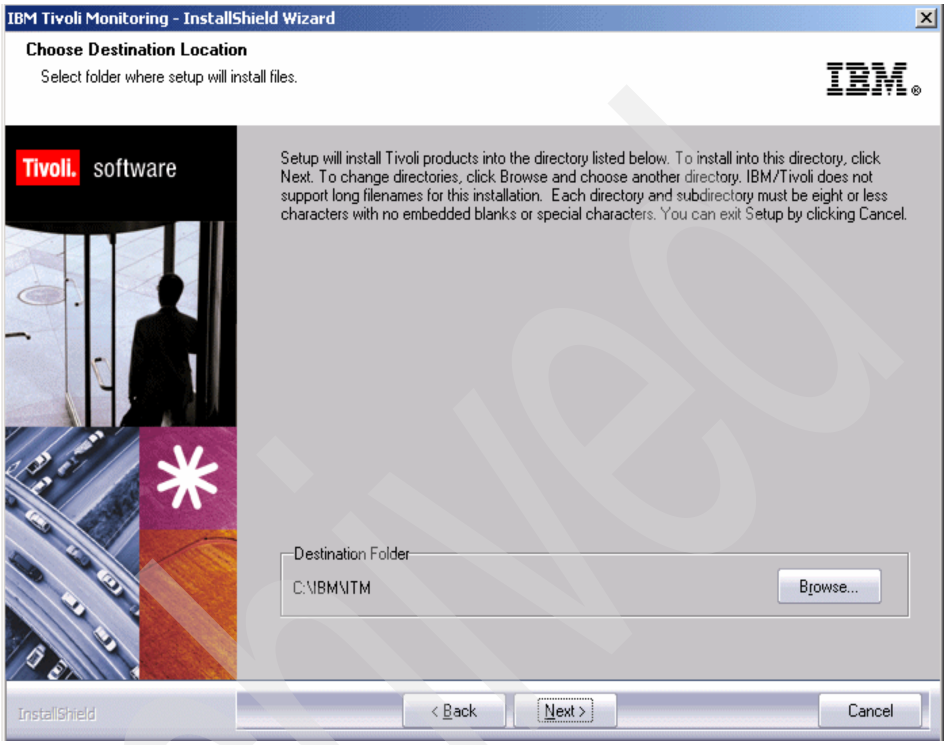

IBM Tivoli Monitoring - InstallShield Wizard
Information
Read the following information and decide.
IBM
Tivoli. software
InstallShield
< Back
Next >
Cancel

Click Next to continue, Cancel to install necessary prerequisite software.

The following list of software MAY be required during this installation.
If required later, selected features may not be available for installation.
If you know the software is required, please stop the installation now and install the prerequisite software.
If you plan to install TEP Server on this computer:
Either DB2 or MS SQL Server is required.
Neither found!

Figure 3-28 IBM Tivoli Monitoring 6.1 requisites information screen

5. Choose the directory where you want to install the product. The default is c:\IBM\ITM as shown in Figure 3-29. Click **Next**.

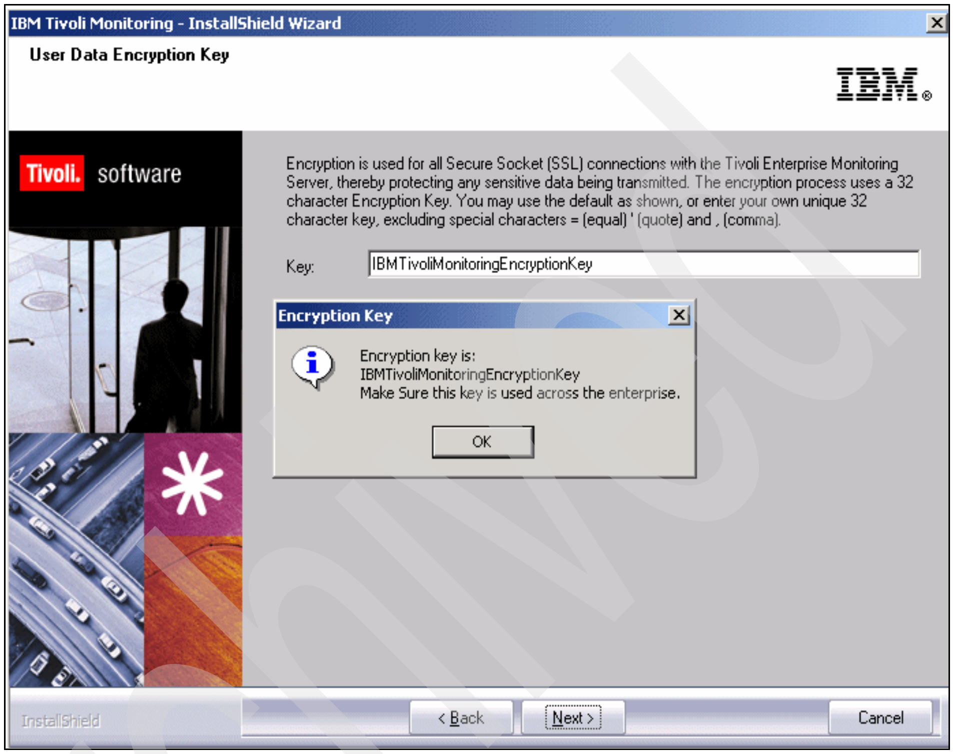

IBM Tivoli Monitoring - InstallShield Wizard
Choose Destination Location
Select folder where setup will install files.
Tivoli. software
Destination Folder
Browse...
InstallShield
< Back
Next >
Cancel

Setup will install Tivoli products into the directory listed below. To install into this directory, click Next. To change directories, click Browse and choose another directory. IBM/Tivoli does not support long filenames for this installation. Each directory and subdirectory must be eight or less characters with no embedded blanks or special characters. You can exit Setup by clicking Cancel.

C:\IBM\ITM

Figure 3-29 IBM Tivoli Monitoring 6.1 default destination installation directory

6. Type the 32-bit encryption key that was used during the installation of the monitoring server to which this monitoring agent connects. Click **Next** and **OK** to confirm the encryption key (Figure 3-30).



Encryption is used for all Secure Socket (SSL) connections with the Tivoli Enterprise Monitoring Server, thereby protecting any sensitive data being transmitted. The encryption process uses a 32 character Encryption Key. You may use the default as shown, or enter your own unique 32 character key, excluding special characters = (equal) ' (quote) and , (comma).

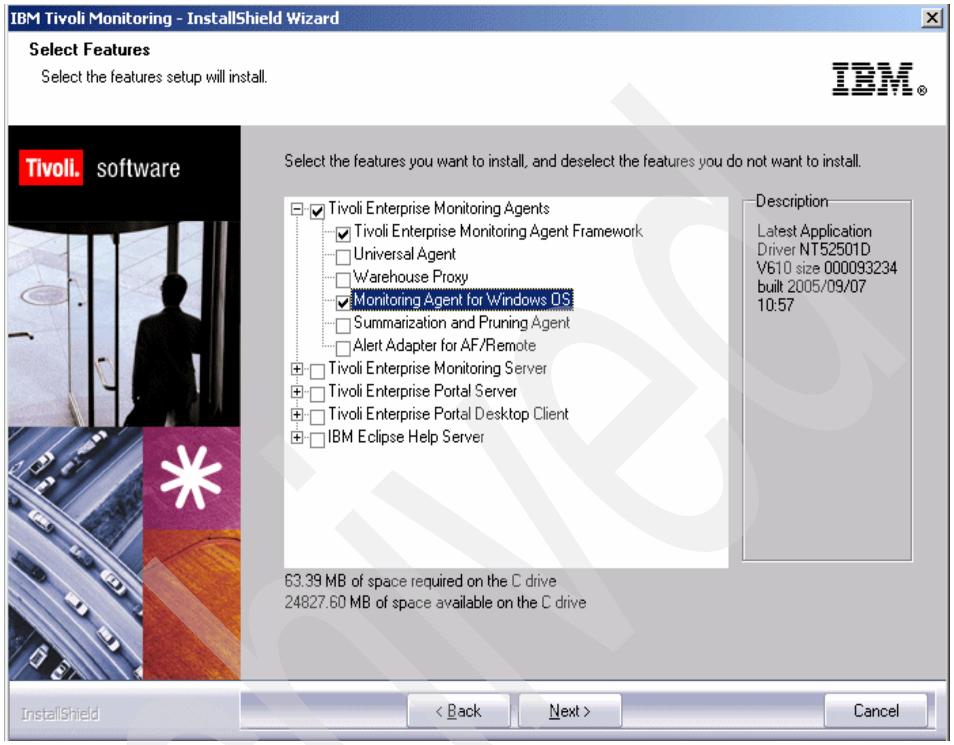
Key: IBMTivoliMonitoringEncryptionKey

Encryption Key

Encryption key is:
IBMTivoliMonitoringEncryptionKey
Make Sure this key is used across the enterprise.

Figure 3-30 IBM Tivoli Monitoring 6.1 encryption key confirmation

7. Expand **Tivoli Enterprise Monitoring Agents** and select the name of the agent that you want to install. Click **Next** (Figure 3-31).


IBM Tivoli Monitoring - InstallShield Wizard

Select Features

Select the features setup will install.

Select the features you want to install, and deselect the features you do not want to install.

- [✓] Tivoli Enterprise Monitoring Agents
  - [✓] Tivoli Enterprise Monitoring Agent Framework
  - [ ] Universal Agent
  - [ ] Warehouse Proxy
  - [✓] Monitoring Agent for Windows OS
  - [ ] Summarization and Pruning Agent
  - [ ] Alert Adapter for AF/Remote
- [ ] Tivoli Enterprise Monitoring Server
- [ ] Tivoli Enterprise Portal Server
- [ ] Tivoli Enterprise Portal Desktop Client
- [ ] IBM Eclipse Help Server

Description

Latest Application Driver NT52501D V610 size 000093234 built 2005/09/07 10:57

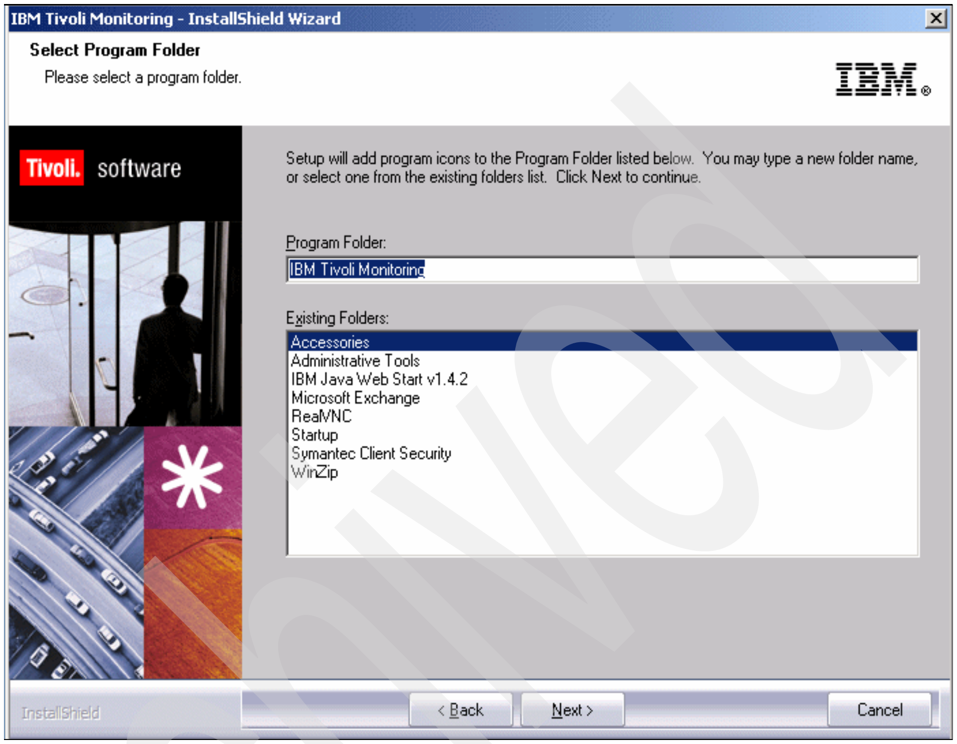
63.39 MB of space required on the C drive

24827.60 MB of space available on the C drive

Figure 3-31 Monitoring agents to be installed

8. Click **Next** on the Agent Deploy window. Do not select any agents.

9. Type a program folder to use in your Start menu and click **Next**. The default folder is IBM Tivoli Monitoring, as shown in Figure 3-32.


IBM Tivoli Monitoring - InstallShield Wizard

Select Program Folder

Please select a program folder.

Setup will add program icons to the Program Folder listed below. You may type a new folder name, or select one from the existing folders list. Click Next to continue.

Program Folder:

IBM Tivoli Monitoring

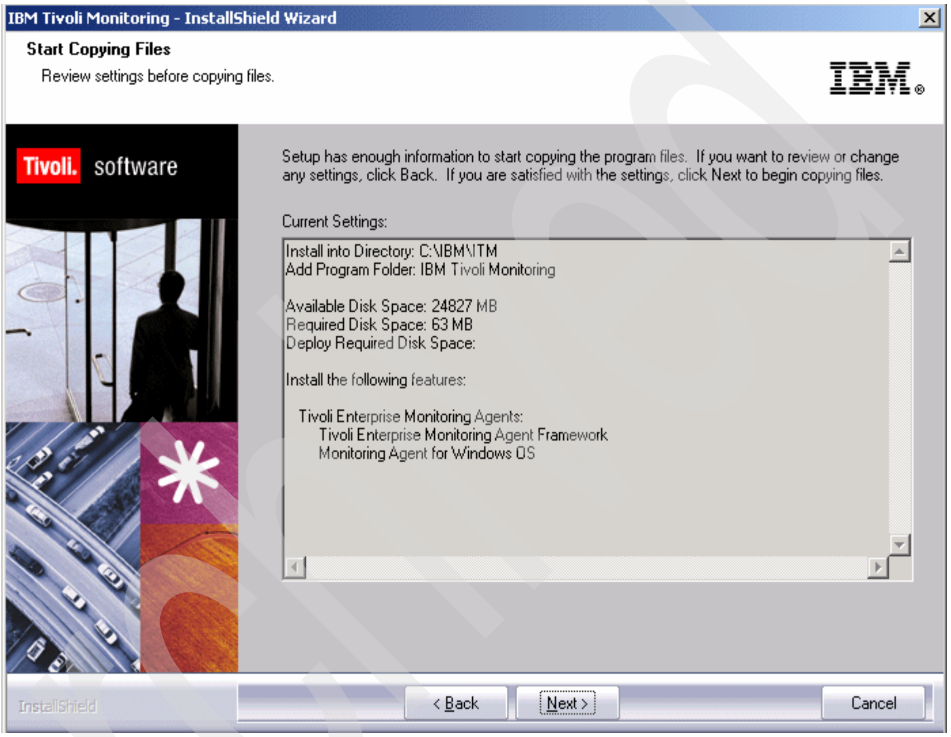
Existing Folders:

Accessories
Administrative Tools
IBM Java Web Start v1.4.2
Microsoft Exchange
RealVNC
Startup
Symantec Client Security
WinZip

Figure 3-32 IBM Tivoli Monitoring 6.1 program folder

10. Review the installation summary details. This summary identifies what you are installing and where you have chosen to install (Figure 3-33). Click **Next** to begin the installation of components.

    After the components are installed and the configuration environment is initialized (indicated by a pop-up window), a configuration window is displayed. Click **Next**.


IBM Tivoli Monitoring - InstallShield Wizard
Start Copying Files
Review settings before copying files.
IBM
Tivoli. software
InstallShield
< Back
Next >
Cancel

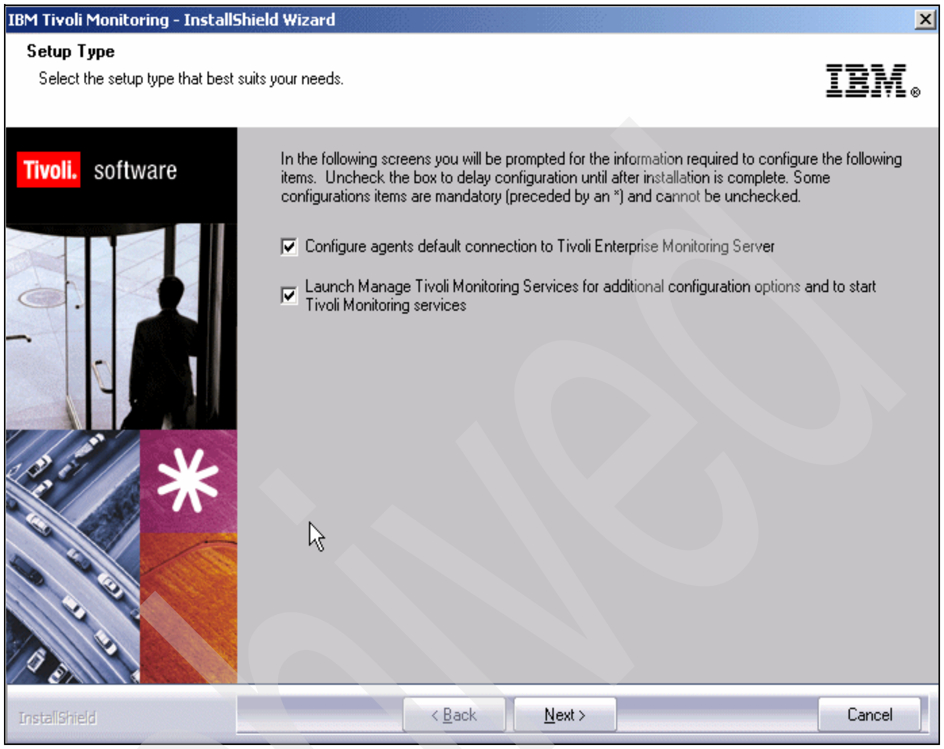
Setup has enough information to start copying the program files. If you want to review or change any settings, click Back. If you are satisfied with the settings, click Next to begin copying files.

Current Settings:

```
Install into Directory: C:\IBM\ITM
Add Program Folder: IBM Tivoli Monitoring

Available Disk Space: 24827 MB
Required Disk Space: 63 MB
Deploy Required Disk Space:

Install the following features:

  Tivoli Enterprise Monitoring Agents:
     Tivoli Enterprise Monitoring Agent Framework
     Monitoring Agent for Windows OS
```

Figure 3-33 Installation summary details

11. Configure the default values for your agent (Figure 3-34). Click **Next**.


IBM Tivoli Monitoring - InstallShield Wizard
Setup Type
Select the setup type that best suits your needs.
IBM
Tivoli. software
InstallShield
< Back
Next >
Cancel

In the following screens you will be prompted for the information required to configure the following items. Uncheck the box to delay configuration until after installation is complete. Some configurations items are mandatory (preceded by an *) and cannot be unchecked.

☑ Configure agents default connection to Tivoli Enterprise Monitoring Server

☑ Launch Manage Tivoli Monitoring Services for additional configuration options and to start Tivoli Monitoring services

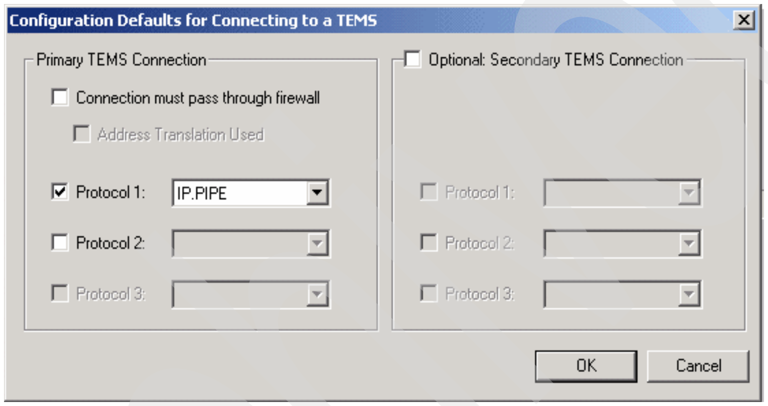
Figure 3-34 Configuration option choice

12. Specify the default values for any IBM Tivoli Monitoring agent to use when they communicate with the monitoring server.

    a. If the agent must cross a firewall to access the monitoring server, select `Connection must pass through firewall`.

    b. Identify the type of protocol that the agent uses to communicate with the monitoring server. Your choices are: IP.UDP, IP.PIPE, IP.SPIPE, or SNA. You can specify three methods for communication, which enables you to set up backup communication methods. If the method you have identified as Protocol 1 fails, Protocol 2 will be used. Click **OK**.

       Figure 3-34 on page 110 shows an example of the protocol communication to be used to communicate with TEMS.


Configuration Defaults for Connecting to a TEMS

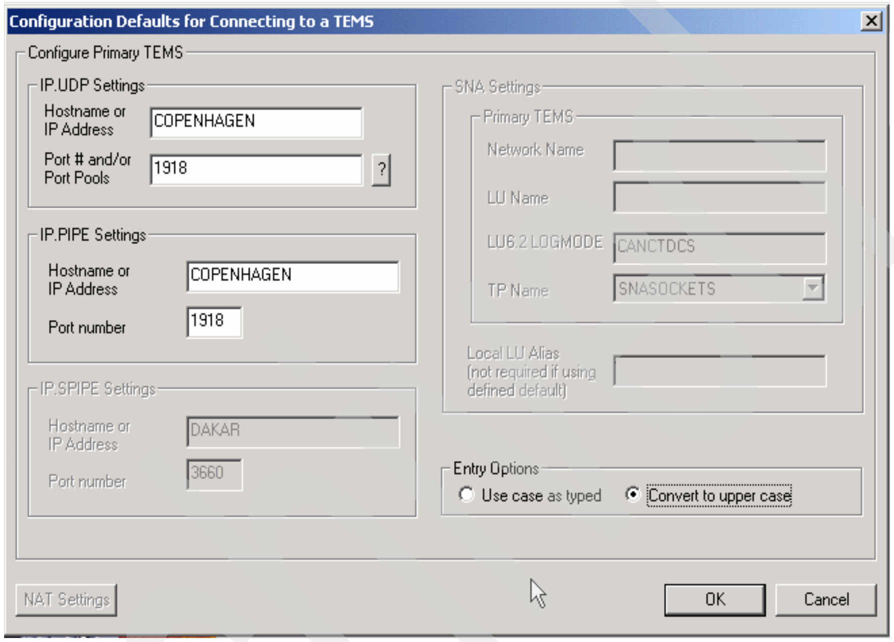
Primary TEMS Connection
- [ ] Connection must pass through firewall
- [ ] Address Translation Used
- [✓] Protocol 1: IP.PIPE
- [ ] Protocol 2:
- [ ] Protocol 3:

[ ] Optional: Secondary TEMS Connection
- [ ] Protocol 1:
- [ ] Protocol 2:
- [ ] Protocol 3:

Figure 3-35 Agent communication protocols

13. Complete the fields to define the communications between agents and the monitoring server. Figure 3-36 shows the primary TEMS the agent will be connected to.

    If you have defined more than one TEMS, a second window will open requesting the configuration of the secondary TEMS host name.



Configuration Defaults for Connecting to a TEMS

Configure Primary TEMS

IP.UDP Settings
Hostname or IP Address: COPENHAGEN
Port # and/or Port Pools: 1918

IP.PIPE Settings
Hostname or IP Address: COPENHAGEN
Port number: 1918

IP.SPIPE Settings
Hostname or IP Address: DAKAR
Port number: 3660

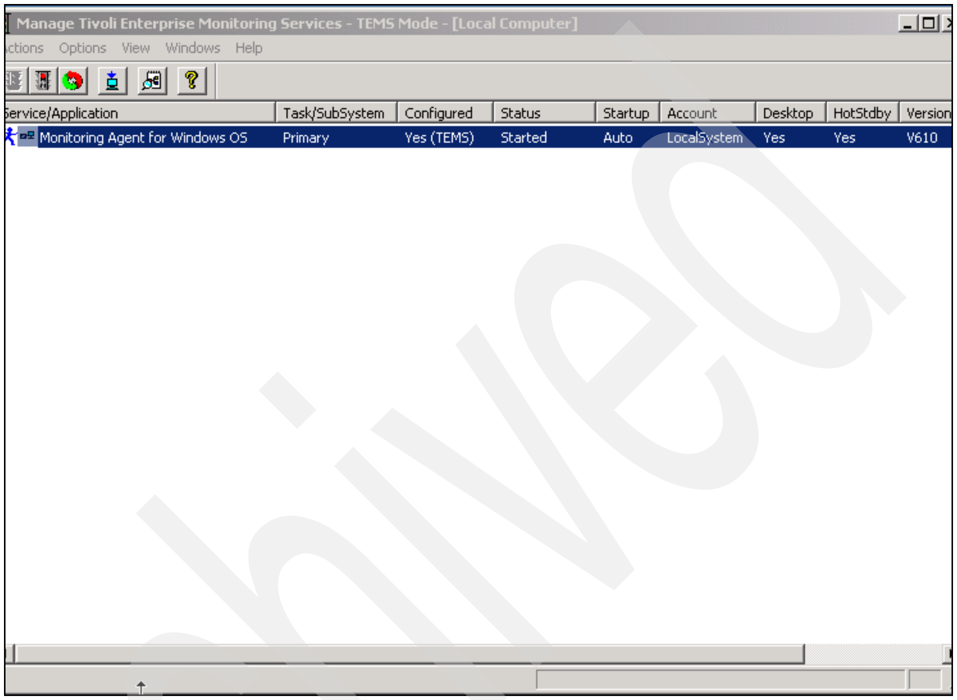
SNA Settings
Primary TEMS
Network Name:
LU Name:
LU6.2 LOGMODE: CANCTDCS
TP Name: SNASOCKETS
Local LU Alias (not required if using defined default):

Entry Options
Use case as typed [✓ Convert to upper case]

Figure 3-36 Agent's TEMS configuration

14. Click **Finish** to complete the installation.

15. Open the **Manage Tivoli Monitoring Services** utility to see whether the monitoring agent has been configured and started as shown in Figure 3-37. If you see `Yes` in the Configured column, the agent has been configured and started during the installation process.



Manage Tivoli Enterprise Monitoring Services - TEMS Mode - [Local Computer]
ctions Options View Windows Help

Service/Application	Task/SubSystem	Configured	Status	Startup	Account	Desktop	HotStdby	Version
Monitoring Agent for Windows OS	Primary	Yes (TEMS)	Started	Auto	LocalSystem	Yes	Yes	V610

Figure 3-37 Tivoli monitoring services console

16. If the value in the Configured column is blank and `Template` is in the Task/Subsystem column:

   a. Right-click the **Template** agent.

   b. Click **Configure Using Defaults**.

   c. Complete any windows requiring information by using the agent-specific configuration settings in the User's Guide for your agent.

   d. Repeat this step as necessary to create monitoring agent instances for each application instance you want to monitor.

The procedure above is used by default to configure any agent from a Windows server.

Installing TEMA on an OS/400 server

Before installing the Monitoring Agent for i5/OS®, complete the following procedures if applicable:

- During installation, you are required to know whether English is the primary language of your iSeries™ system. To determine this, complete the procedure in the next section, “Determining the primary language of your iSeries system.”
- If you are using TCP/IP for network communications, verify that your TCP/IP network services are configured to return the fully qualified host name of the computer where you will install the monitoring agent as described in “Verifying the TCP/IP configuration” on page 6.
- If you have a previous version of a Candle Monitoring Agent installed, delete it as in “Deleting previous versions of the monitoring agent” on page 7.

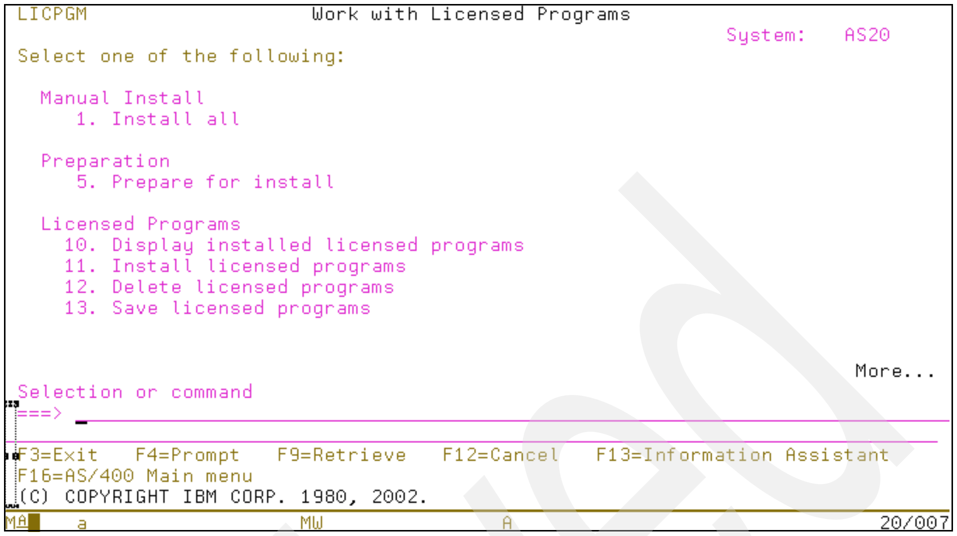
Determining the primary language of your iSeries system

Use the following procedure to determine the primary language of your iSeries system:

1. Log on onto your system as user QSECOFR.
2. From an i5/OS command line, enter this command:

   ```
   GO LICPGM
   ```

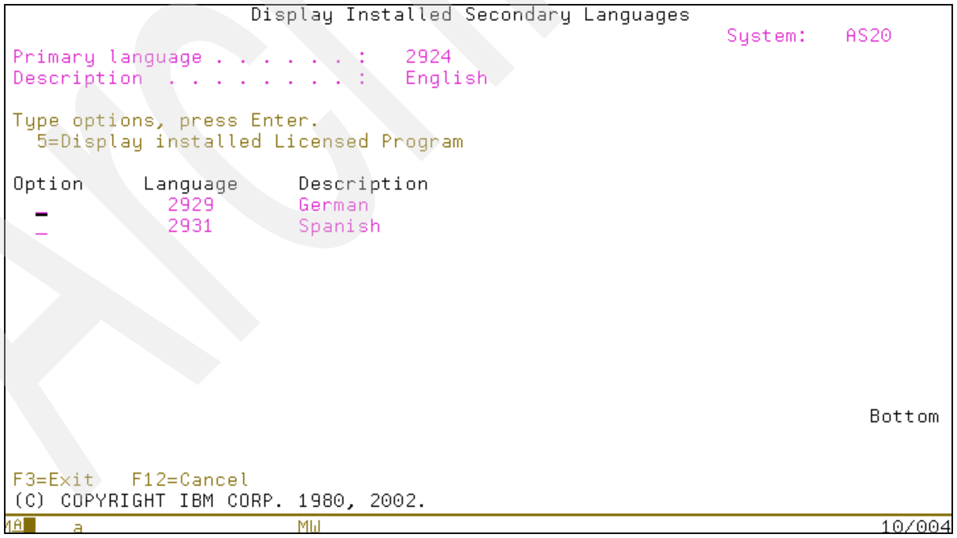
   See Figure 3-38 on page 115.



```
LICPGM                      Work with Licensed Programs
                                                              System:   AS20
Select one of the following:

  Manual Install
     1. Install all

  Preparation
     5. Prepare for install

  Licensed Programs
    10. Display installed licensed programs
    11. Install licensed programs
    12. Delete licensed programs
    13. Save licensed programs

                                                                      More...
Selection or command
===> _

F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel   F13=Information Assistant
F16=AS/400 Main menu
(C) COPYRIGHT IBM CORP. 1980, 2002.
MA      a                    MW                 A                     20/007
```

Figure 3-38   Main OS/400 Menu window

3. Enter 20 (Display installed secondary languages).
4. Note the primary language and description that is displayed in the upper-left corner of the window. For an English language system, the primary language is 2924, and the description is English. See example on Figure 3-39.



```
                         Display Installed Secondary Languages
                                                              System:   AS20
Primary language . . . . . . . :    2924
Description  . . . . . . . . . :    English

Type options, press Enter.
  5=Display installed Licensed Program

Option      Language      Description
  _           2929        German
  _           2931        Spanish

                                                                       Bottom

F3=Exit   F12=Cancel
(C) COPYRIGHT IBM CORP. 1980, 2002.
MA      a                    MW                                       10/004
```

Figure 3-39   Primary language OS/400 definition window

Verifying the TCP/IP configuration

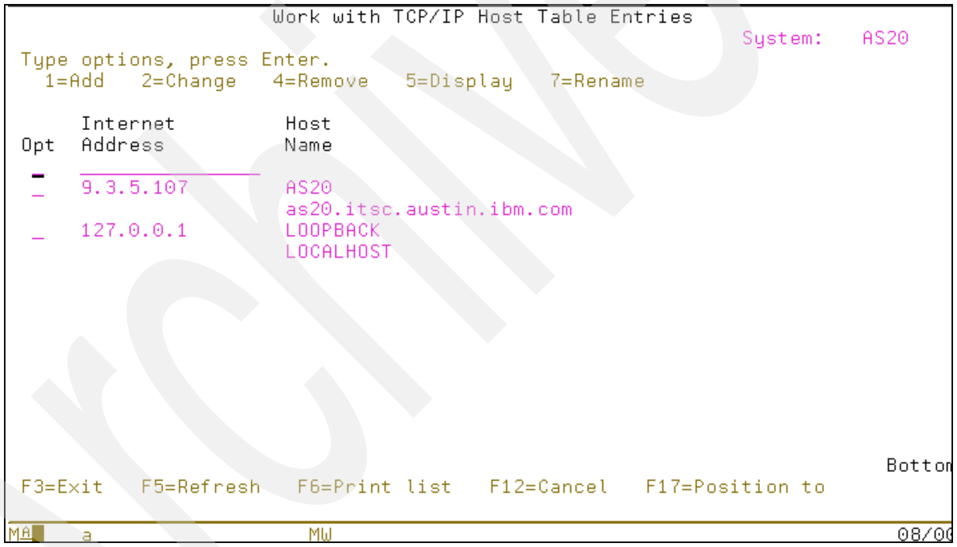
Ensure that your TCP/IP network services are configured to return the fully qualified host name (for example, myhost.ibm.com). The following procedure shows how to check whether TCP/IP is properly configured on your system. This procedure is not necessary if you are using SNA for network communications.

Required authorization role is *IOSYSCFG.

1. From an i5/OS command line, enter the following command:

   ```
   CFGTCP
   ```

2. Select **Work with TCP/IP host tables entries: option 10**.
3. Confirm that the first entry in the Host Name column is the fully qualified host name that is associated with the IP address of the iSeries computer where you plan to install the monitoring agent (Figure 3-40). If it is not, change the entry to the fully qualified host name.



```
                      Work with TCP/IP Host Table Entries
                                                            System:   AS20
Type options, press Enter.
  1=Add   2=Change   4=Remove   5=Display   7=Rename

     Internet           Host
Opt  Address            Name
 _
 _   9.3.5.107          AS20
                        as20.itsc.austin.ibm.com
 _   127.0.0.1          LOOPBACK
                        LOCALHOST

                                                                        Botto
F3=Exit   F5=Refresh   F6=Print list   F12=Cancel   F17=Position to

MA     a                  MW                                              08/0
```

Figure 3-40   OS/400 TCP/IP configuration panel

4. Return to the Configure TCP/IP menu and select **Change TCP/IP domain information**; option 12.
5. Confirm that a host name and domain name are provided and that they match the entry you just confirmed in the TCP/IP Host Table.
6. Confirm that the first entry for Host name search priority is `*LOCAL`.

Installing the monitoring agent

You can install the Monitoring Agent for i5/OS from a PC or from an iSeries computer, whichever method is more convenient at your site. This procedure includes instructions for both methods.

Required authorization role “Sign on as QSECOFR or with a profile with an equivalent special authority (SPCAUT)”:

- *ALLOBJ
- *AUDIT
- *IOSYSCFG
- *JOBCTL
- *SAVSYS
- *SECADM
- *SERVICE
- *SPLCTL

Note: Before beginning this procedure, install IBM Tivoli Monitoring and the Tivoli Enterprise Portal.

When you finish Configure the Monitoring Agent for i5/OS as described in the previous sections, run the following procedure:

1. From an i5/OS command line, ensure that the QALWOBJRST system value is set to *ALL. To do this, follow these steps:
   a. Enter the following command:
      ```
      WRKSYSVAL QALWOBJRST
      ```
   b. Select **5** (Display) and verify that the value is set to *ALL.
   c. Press Enter to continue.
   d. If the value of QALWOBJRST is set to *ALL, skip to step 3.
      If the value of QALWOBJRST is not set to *ALL, make note of the values and go to step 2.
2. If the value of QALWOBJRST is not set to *ALL, follow these steps:
   a. On the Work with System Values window, enter `2` to change the values.
   b. On the Change System Value window, change the existing values to `*ALL` and press Enter.
   c. Press F3.
3. From an i5/OS command line, enter the following command to create an i5/OS CCCINST library for the Monitoring Agent for i5/OS installation if this library does not already exist:
   ```
   CRTLIB LIB(CCCINST)
   ```

4. Enter the following command to create a save file in the CCCINST library for the Monitoring Agent for i5/OS:

   ```
   CRTSAVF CCCINST/A4520CMA TEXT('ITM 61 i5/OS')
   ```

   Note: When pasting this command to an iSeries session, the single-quote (') characters that enclose the text string might be missing. If this happens, manually add the single-quote characters for the command to work.

5. Transfer the software for the Monitoring Agent for i5/OS to the target iSeries computer.

   On a Windows PC, follow these steps:

   a. Insert the IBM Tivoli Monitoring 6.1 product CD into the PC CD-ROM drive.
   b. Enter the following command to create a work folder:

      ```
      WRKFLR
      ```

   c. Select **1 (Create Folder)** and specify the following name for the folder:

      ```
      A4FLR
      ```

   d. Enter the following command:

      ```
      WRKLNK QOPT
      ```

      The Work with Object Links window displays the qopt object link.
   e. Select **5 (Next Level)** at the qopt object link to select the next object link: the volume ID of the CD-ROM. Make note of this volume ID for use in the remainder of this procedure.
   f. Continue to select **5** for each link level until the path /QOPT/*volume_id*/OS400/TMAITM6 is displayed, where *volume_id* is the volume ID of the CD-ROM drive from step e.
   g. Look for the A4520CMA.SAV file and enter the following command to copy this save file to the QDLS directory:

      ```
      CPY OBJ('/QOPT/volume_id/OS400/TMAITM6/A4520CMA.SAV')
      TODIR('/QDLS/A4FLR')
      ```

      where *volume_id* is the volume ID of the CD-ROM drive from step 5e.
   h. Enter the following command to start an FTP session:

      ```
      ftp computer_name
      ```

      computer_name is the name of the target i5/OS computer.
   i. Enter the following command to change to the file type to binary:

      ```
      binary
      ```

j. Enter the following command:

```
NAMEFMT 1
```

k. Enter the following command to transfer the software for the monitoring agent:

```
put /QDLS/A4FLR/A4520CMA.SAV /QSYS.LIB/CCCINST.LIB/A4520CMA.SAVF
```

l. Press F3 and select **1** to end the FTP session.

6. From an i5/OS command line, install the software for the Monitoring Agent for i5/OS:

   – If you are installing the monitoring agent on a system that is set to the English language (language ID 2924), enter the following command:

   ```
   RSTLICPGM LICPGM(0KA4610) DEV(*SAVF) SAVF(CCCINST/A4520CMA)
   ```

   – If you are installing the monitoring agent on a system that is not set to language ID 2924, enter the following two commands:

   ```
   RSTLICPGM LICPGM(0KA4610) DEV(*SAVF) RSTOBJ(*PGM)
   SAVF(CCCINST/A4520CMA)

   RSTLICPGM LICPGM(0KA4610) DEV(*SAVF) RSTOBJ(*LNG) LNG(2924)
   SAVF(CCCINST/A4520CMA) LNGLIB(QKA4LNG)
   ```

7. If you plan to install other monitoring agents, leave the value of QALWOBJRST set to *ALL until you are finished. If you do not plan to install other monitoring agents, change the value of QALWOBJRST to the values you recorded in 1d on page 117.

8. Optional: Enter the following command to delete the installation library, which is no longer needed:

   ```
   DLTLIB CCCINST
   ```

9. Optional for an iSeries computer: Delete the A4520CMA.SAV file from your folder. Follow these steps:

   a. Enter the following command:

   ```
   WRKDOC FLR(A4FLR)
   ```

   b. Enter 4 for the A4520CMA.SAV file.

   c. Press Enter to return to the command line.

   d. Enter the following command to delete the installation folder:

   ```
   WRKFLR
   ```

   e. Enter 4 for the A4FLR folder.

   f. Press F3 to return to the command line.

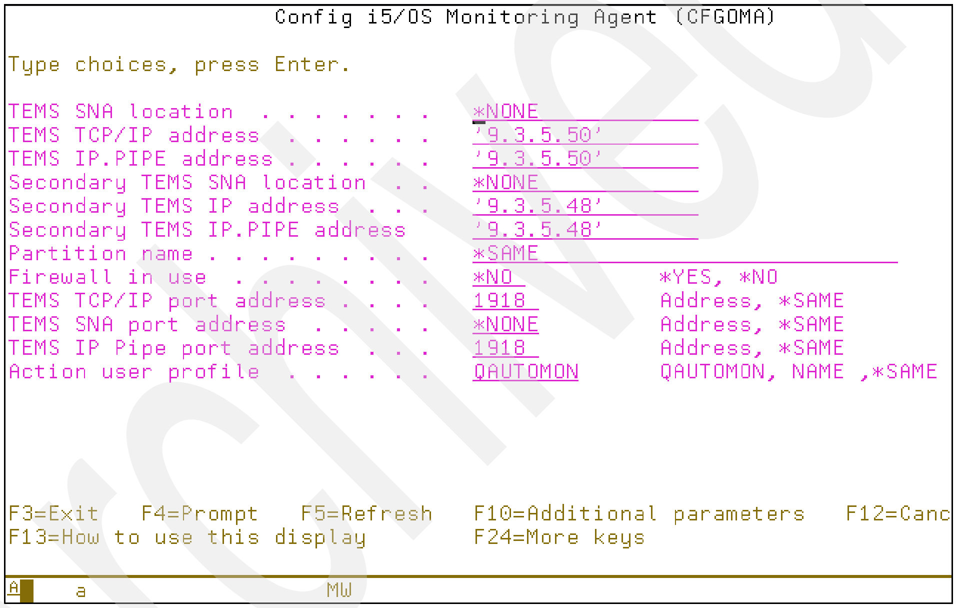
Configuring the Monitoring Agent for i5/OS

Use the following procedure to configure or reconfigure the network connections between the Monitoring Agent for i5/OS and the Tivoli Enterprise Monitoring Server (monitoring server).

1. From an i5/OS command line, enter the following command:

```
GO OMA
```

2. Enter **4** (Configure Tivoli Monitoring: i5/OS Agent). The Config i5/OS Monitoring Agent (CFGOMA) window appears.
3. Enter your site's values for the configuration parameters as shown in Figure 3-41.



```
                    Config i5/OS Monitoring Agent (CFGOMA)

Type choices, press Enter.

TEMS SNA location  . . . . . . .   *NONE
TEMS TCP/IP address  . . . . . .   '9.3.5.50'
TEMS IP.PIPE address . . . . . .   '9.3.5.50'
Secondary TEMS SNA location  . .   *NONE
Secondary TEMS IP address  . . .   '9.3.5.48'
Secondary TEMS IP.PIPE address     '9.3.5.48'
Partition name . . . . . . . . .   *SAME
Firewall in use  . . . . . . . .   *NO           *YES, *NO
TEMS TCP/IP port address . . . .   1918          Address, *SAME
TEMS SNA port address  . . . . .   *NONE         Address, *SAME
TEMS IP Pipe port address  . . .   1918          Address, *SAME
Action user profile  . . . . . .   QAUTOMON      QAUTOMON, NAME ,*SAME




F3=Exit   F4=Prompt   F5=Refresh   F10=Additional parameters   F12=Canc
F13=How to use this display        F24=More keys
A    a                   MW
```

Figure 3-41 Configuring the monitoring agent

4. Optional: Customize the data collection intervals by changing the values of the following configuration variables in the KMSPARM(KBBEV) file, which are listed with their default values:
   – KA4_JOB_DATA_INTERVAL=15
   – KA4_IOP_DATA_INTERVAL=30
   – KA4_DISK_DATA_INTERVAL=30
   – KA4_POOL_DATA_INTERVAL=15
   – KA4_COMM_DATA_INTERVAL=60

Valid values for these configuration variables are 15, 30, 60, 120, and 240. These configuration variables follow the rules of the collection interval parameter of the i5/OS QPMWKCOL API. Keep the following items in mind:

– Disk and IOP-related data require a minimum of 30 seconds between collection intervals.

– Communication-related data requires a minimum of 60 seconds between collection intervals.

– Collect job-related data as infrequently as possible to minimize the impact on system performance.

– The i5/OS collection services performance data collector supports data collection at one-minute intervals, not at two-minute or four-minute intervals. Therefore, when using the API and requesting data at two-minute or four-minute intervals, the data is collected at one-minute intervals, but reported back every two or four minutes.

5. Optional: Customize the time interval for custom situations by changing the value of the following configuration variable in the KMSPARM(KBBEV) file, which is listed with its default value:

```
KA4_COMM_SIT_INTERVAL=3600
```

Note: Custom situations created using APPN Topology or communication attributes use the time interval specified by the KA4_COMM_SIT_INTERVAL configuration variable. Change the value of this configuration variable if you want communication-related alerts to be raised at a different interval. The default interval is set to 3600 to prevent overloading the monitoring agent, because these custom situations are also created as a workaround for a known problem of agent failure when running consecutive APPN reports or running a situation after running an APPN report. These situations might be continuously running and based on events and can overload the monitoring agent when alerting for these conditions.

Starting the monitoring agent

The following steps show how to start the Monitoring Agent for i5/OS.

Background information

When the Monitoring Agent for i5/OS is started, you can use the associated CLI commands. Table 3-7 on page 122 shows the group profiles that are authorized to these commands by default when the Monitoring Agent for i5/OS is first installed. A check mark in a column indicates that users associated with that group profile can use the command.

*Table 3-7 Commands owned by QSYS with *PUBLIC *CHANGE*

Command	QSRV	QSRVBAS	QSYSOPR	QPGMR
CFGOMA	√			
DSPOMALOG	√	√	√	√
ENDOMA	√		√	
STROMA	√		√	

To determine which group profile a user is associated with, use this command:

```
Display User Profile (DSPUSRPRF)
```

The group profile to which the user is associated is listed in the group profile field.

Required authorization role

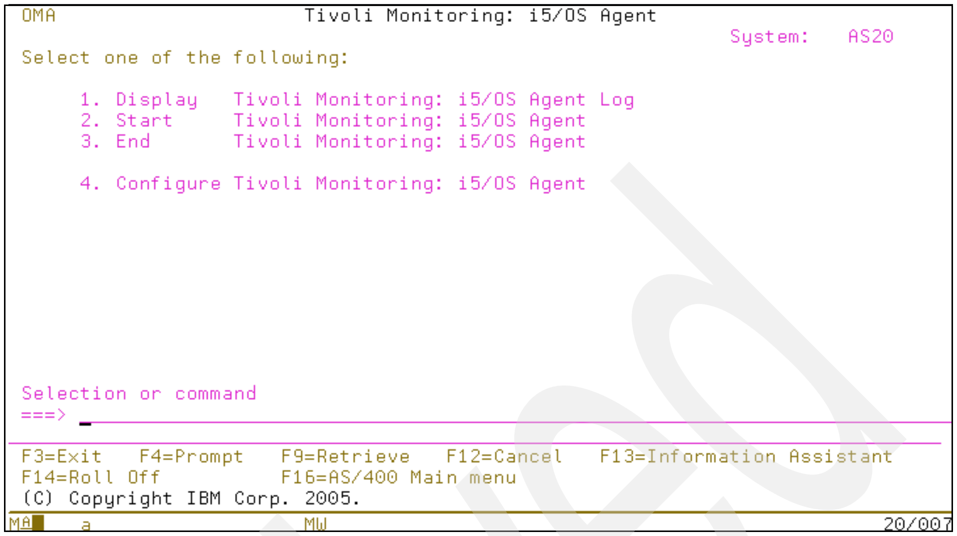
*USER or, in some cases, *JOBCTL special authority if authorities for QAUTOMON were changed after installation.

Starting the agent

1. From an i5/OS, enter the following command:

   ```
   GO OMA
   ```

   This opens the window in Figure 3-42 on page 123.



```
 OMA                      Tivoli Monitoring: i5/OS Agent
                                                                System:   AS20
 Select one of the following:

      1. Display  Tivoli Monitoring: i5/OS Agent Log
      2. Start    Tivoli Monitoring: i5/OS Agent
      3. End      Tivoli Monitoring: i5/OS Agent

      4. Configure Tivoli Monitoring: i5/OS Agent









 Selection or command
 ===> _

 F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel   F13=Information Assistant
 F14=Roll Off          F16=AS/400 Main menu
 (C) Copyright IBM Corp. 2005.
MA    a                        MW                                          20/007
```

Figure 3-42 Monitoring agent configuration window

2. Enter 2 (Start Tivoli Monitoring: i5/OS Agent). The greater-than character (>) preceding option 2 indicates that the monitoring agent is not started. When the monitoring agent is started the greater-character (>) is not displayed.

Important: If you did not seed the hub and Remote TEMS during their installation with this type of application (i5/OS), you must do it before the agent can be up and running properly. By default i5/OS is selected for seeding during TEMS installation. You can verify the supported application by using the `cinfo` command on the Hub and Remote TEMS.

Note: Use the same procedure to stop the monitoring agent, selecting the option 3.

Deploying TEMA from TEPS

To deploy a monitoring agent through the portal, you first have to deploy the OS monitoring agent on the system. Then you will be able to deploy any other agent from the TEP.

Note: You also must install the bundles on the server where the monitoring agent is being deployed.

Use the following steps to deploy an agent through the portal GUI:

1. Open the Tivoli Enterprise Portal.
2. In the Navigation tree, navigate to the computer where you want to deploy the agent.
3. Right-click the computer and click **Add Managed System**.
4. Select the agent that you want to deploy and click **OK**.
5. Select the configuration fields required for the agent. For information about these fields, see the configuration documentation for the agent that you are deploying.
6. Click **Finish**.
7. If the computer where you are deploying the agent already has a version of that agent installed, you can either stop the deployment or, if the existing version is older than the version you are deploying, specify to replace the existing agent with this new agent. To replace the existing agent, click **Yes**.
8. Click **Finish** on the message that tells you that deployment was successful.

3.2.9 Deploying TEMA from command line interface

In many cases you must create a node (install an OS agent) from the command line. Before beginning the procedure, be sure that the packages are already in the depot server where you will execute the commands.

Deploying a Windows OS agent from a Remote TEMS using command line

1. Check whether the target platform bundles are already installed in the server. Execute the following command:

   ```
   tacmd login -s tems_hostname -u tems_user -p tems_password
   ```

   In this command:

   – *tems_hostame* is the host name where you want to initiate the installation.

   – *tems_user* is the user ID on the tems_hostname.

   – *tems_password* is the tems_hostname password.

2. Execute the following command:

   ```
   tacmd listBundles
   ```

3. If the bundles are not present, execute the following commands to add them on the server:

   ```
   cd image_dir
   ```

image_dir is the directory where you untar the IBM Tivoli Monitoring 6.1 codes.

```
tacmd addBundles -i bundles_path
```

4. Deploy the agent on the targeted server executing the following command:

```
tacmd createNode -h server -u user -w password -d target_directory
```

Table 3-8 on page 148 installs the Window agent on london from copenhagen using the -p option to set the agent properties.

Example 3-8 Deploying the agent on the targeted server

```
C:\>tacmd createNode  -h london -u Administrator -w **** -d c:/IBM/ITM -p
KEY=IBMTivoliMonitoringEncryptionKey PROTOCOL1=IP.PIPE PROTOCOL2=IP.UDP
PORT=1918 SERVER=COPENHAGEN BSERVER=EDINBURG BPROTOCOL1=IP.UDP
PROTOCOL2=IP.PIPE PORT=1918
KUICCN001I Initializing required services...
KUICCN039I Attempting to connect to host london ...
KUICCN050I Distributing file 203 of 203 (85.9 MB / 85.9 MB)...
KUICCN002I Beginning the installation and configuration process...

KUICCN057I The node creation on host london was successful.

KUICCN065I The node creation operation was a success.
```

Deploying an application agent from a Remote TEMS using command line

The same procedure can be used to install a monitored agent using the addSystem option instead of createNode. The following command performs this operation:

```
tacmd addSystem -t type -n node_name -p SECTION.NAME=value ....
```

Example 3-9 shows how to deploy a Microsoft SQL Server agent on a nice server to monitor the wproxy database instance.

Example 3-9 Deploying an application agent

```
tacmd addSystem -t OQ -n Primary:nice:NT -p DBSETTINGS.db_sid=MyServer
DBSETTINGS.db_login=sa DBSETTINGS.db_password=sapwd
"DBSETTINGS.db_home=c:\Program Files\Microsoft SQL Server\MSSQL"
"DBSETTINGS.db_errorlog=C:\Program Files\Microsoft SQL
Server\MSSQL\LOG\ERRORLOG" INSTANCE=wproxy
```

Starting the TEMA from the Remote TEMS

After the agent is installed, start it by executing the following procedure:

1. Log on to the agent Remote TEMS by executing the following command:

   ```
   tacmd login -s remote_tems -u user -p password
   ```

2. When you are logged in, execute the command to start the agent:

   ```
   tacmd startAgent -n hostname -t pc
   ```

   pc is the product code (*lz* for Linux, *nt* for Windows, *um* for Universal Agent).

3.2.10 Installing a new managed system: Microsoft Exchange example

We use the Microsoft Exchange Server deployment as an example of deploying a new managed system on a TEMS environment.

Installing the OS Agent

First, install the OS Agent on the Microsoft Exchange Server. (Refer to "Tivoli Enterprise Monitoring Agent" on page 95.) You can install the OS Agent either locally or remotely using **`tacmd createNode`** on the command line.

Installing the application agent

After you install the OS Agent, install the Microsoft Exchange Server Agent. Refer to the procedure in "Installing TEMA on a Windows server" on page 102 if you want to install the agent locally, or "Deploying TEMA from TEPS" on page 123 if you want to deploy the agent from the TEPS. If you want to deploy the agent from a TEMS command line interface, refer to "Deploying an application agent from a Remote TEMS using command line" on page 125.

Installing support for the agent being deployed

Install the support on the Hub TEMS, the Remote TEMS, and the TEPS.

Note: If you fail to install the support (seed) for the agent, you will be able to see the agent on the TEP, but you will not see any data when you click on the attribute group, only a message saying that the request has failed.

3.2.11 Tivoli Enterprise Portal (TEP)

There are two ways to access the Tivoli Enterprise Portal: browser and desktop client. The browser-based client has two main advantages: There is no need to install an updated client if a newer version is available; the browser client will always be at the latest level available from the server. Also, you can store links to some of your favorite workspaces as you would store any other link in a browser.

The only downside to the browser-based client is the fact that you lose desktop real estate taken up by the browser's headers. From a functional point of view, there are no differences between the two.

Tip: IBM Tivoli Monitoring requires IBM Java V 1.4.2 to be installed on all systems where the TEP browser client will be initiated. One likely error that signifies this is: `KFWITM215E - unable to process login request.`

Installing Tivoli desktop client on a Windows machine

Use the following steps to install the desktop client for Tivoli Enterprise Portal:

1. On the computer where you want to install the desktop client, start the installation wizard by launching the **setup.exe** file from the installation media.
2. Click **Next** on the welcome window.
3. Accept the software license by clicking **Accept**.
4. Read the information regarding potentially missing prerequisites and click **Next**.
5. Specify the directory where you want to install the portal software and accompanying files. The default location is C:\IBM\ITM. Click **Next**.
6. Type an encryption key to use. This key should be the same as the one that was used during the installation of the portal server to which the client will connect. Click **Next** then **OK** to confirm the encryption key.
7. Select **Tivoli Enterprise Portal client**.
8. If you want to view IBM Tivoli Enterprise Console events through the Tivoli Enterprise Portal, expand **Tivoli Enterprise Portal client** and ensure that **Tivoli Enterprise Console GUI Integration** is selected.
9. Click **Next**.
10. Click **Next** without selecting any agents to deploy.
11. Specify the program folder name and click **Next**.
12. Confirm the installation details and click **Next** to start the installation. After the installation is complete, a configuration window is displayed.
13. Click **Next** to configure the connection to the portal server, the connection to the monitoring server, and to launch Manage Tivoli Monitoring Services.
14. Type the host name of the portal server and click **OK**.
15. Configure the default connection to the monitoring server:
    a. If the agent must cross a firewall to access the monitoring server, select **Connection must pass through firewall**.

   b. Identify the type of protocol that the agent uses to communicate with the hub monitoring server. Click **OK**.

16. Complete the rest of the fields (refer to Table 3-5 on page 77) for the monitoring server, and click **Finish** to complete your installation.

Installing the desktop client on a Linux machine

Use the following steps to install and configure the portal desktop client on a Linux computer.

Installing the desktop client

Use the following steps to install the portal server and desktop client:

1. In the directory where you extracted the installation files, run the following command:

   ```
   ./install.sh
   ```

2. When prompted for the IBM Tivoli Monitoring home directory, press Enter to accept the default (opt/IBM/ITM) or change the directory as per your needs.
3. Type `y` to create this directory when prompted. If the directory already exists, you will receive the following type of message:

   ```
   CANDLEHOME directory "/opt/IBM/ITM" already exists.
   OK to use it [ y or n; "y" is default ]?  y
   ```

   Select option `1` when the message in Example 3-10 is displayed.

Example 3-10 Selecting install options

```
Select one of the following:
1) Install products via command line.
2) Install products to depot via command line.
3) Exit install.
Please enter a valid number:
```

4. Type 1.
5. Type the number that corresponds to the language in which you want to display the software license agreement and press Enter.
6. Press Enter to display the agreement.
7. Type `1` to accept the agreement and press Enter.
8. Type the encryption key that was used during the installation of the portal server to which the client will connect, and press Enter. This displays a numbered list of available operating systems. If you have already installed

other IBM Tivoli Monitoring 6.1 components on that machine, the key should already be defined and you will receive the following message

```
runGSkit        :  Keyfile.kdb already exists, skipping keyfile and
certificate creation.
```

9. Press Enter to accept the OS type; the default value is your current operating system. If not, type the number for the operating system that you are installing on.
10. Type `y` to confirm the operating system and press Enter. This displays a numbered list of available components.
11. A list of products are presented as shown in Example 3-11. Type the number corresponding to the TEP Desktop client: `6`.

Example 3-11 List of products

```
The following products are available for installation:

 1) IBM Eclipse Help Server V610R104
 2) Monitoring Agent for Linux OS V610R115
 3) Monitoring Agent for UNIX Logs V610R121
 4)Summarization and Pruning agent V610R141
 5) Tivoli Enterprise Monitoring Server V610R215
 6) Tivoli Enterprise Portal Desktop Client V610R172
 7) Tivoli Enterprise Portal Server V610R172
 8) Tivoli Enterprise Services User Interface V610R194
 9) Universal Agent V610R229
10) all of the above
```

12. Type `y` to confirm the installation. The installation begins.
13. After all of the components are installed, you are asked whether you want to install components for a different operating system. Type `n` and press Enter. Installation is complete. You will see the message in Example 3-12.

Example 3-12 Installation complete message

```
Installation step complete.

You must install TEMS support for the agent products.
This is done by starting and seeding the TEMS for the supported agents.

You may now reconfigure any installed IBM Tivoli Monitoring product via the
"/opt/IBM/ITM/bin/itmcmd config" command.
```

The next step is to configure the TEP desktop client on the server.

Configuring the portal desktop client on Linux

Use the following steps to configure the desktop client on Linux:

1. At the command line change to the opt/IBM/ITM/bin directory.
2. Run the following command:

   ```
   ./CandleConfig -A cj
   ```

3. Type your TEP instance and press Enter, or just press Enter to use the default instance name. We use `tep_ankara`, ankara being the server name where we are installing TEP.
4. Type the host name for the portal server and press Enter.
5. Type your browser path directory and press Enter.
6. Press Enter when you are asked whether you want to use HTTP Proxy support. The default value is no. The desktop client is now configured. The next step is to start the portal server and portal desktop client.

   Example 3-13 shows the TEP desktop configuration on server ankara.

Example 3-13 TEP desktop configuration

```
[root@ankara bin]# ./CandleConfig -A cj
CandleConfig       :  installer level  400 / 100.
CandleConfig       :  running li6243 jre.
Agent configuration started...
Enter TEP Instance Name(Default is: none): tep_ankara
TEP Server Hostname(Default is: none): berlin
Browser Path(Default is: /usr/bin/mozilla):
HTTP Proxy Support? [YES or NO] (Default is: no):
Agent configuration completed...
```

7. Seed the TEMS with the following command line:

   ```
   ./CandleSeed -t tems_name pc
   ```

   where pc = cj

8. Start the agent executing the following command:

   ```
   ./itmcmd agent start cj
   ```

3.2.12 Warehouse Proxy installation and configuration

This section describes the steps for installing and configuring a Warehouse Proxy agent. The Tivoli Data Warehouse needs a relational database to store the historical data. DB2 UDB is the preferred database, but MS SQL and Oracle are also supported. The product ships with a copy of DB2 UDB.

Preinstallation configuration

Before initiating the Warehouse Proxy agent installation and configuration, you must create a Windows user.

Note: ITMUser is default user used in the Warehouse Proxy agent.

Creating a Windows user

Use the following steps to create a Windows user called ITMUser; remember that the database used for this book is DB2 so if you use another database, the configuration can be different:

1. Open the Computer Management window.
2. In the navigation pane of the Computer Management window, expand **Local Users** and **Groups**.
3. Right-click the **Users** folder and click **New User**.
4. Type `ITMUser` in the User Name field.
5. Type `marath0n` in the Password field. Type the password again in the Confirm password field to confirm it.

   Note: You can set a different user and password if you prefer.

6. Clear the **User must change password at next logon** check box.
7. Click **Create** to create the user.
8. Click **Close** to close the window.
9. Click the **Groups** folder.
10. Double-click **DB2ADMNS** in the right pane of the window.

    Note: DB2ADMNS group is created with the DB2 database installation.

11. Perform the following step if you are using Windows 2003:
    a. Click **Add**.
    b. Type `ITMUser` in the Enter the names to select (examples) field and click **Check Names**.
    c. Click **OK** and then click **OK** again.

12. Perform the following step if you are using Windows 2000:
    a. Click **Add** in the Administrator Properties window.
    b. Locate **ITMUser**, the new user you created, and select it.
    c. Click **Add**.
13. Click **OK** twice to close the Administrator Properties window.
14. Close the Computer Management window.

Important:

1. You must create a database for the IBM Tivoli Monitoring Warehouse proxy if you are using a non-DB2 database or you are using DB2 database on a UNIX server. You also must create an ODBC that points to the database you created to store the data in the datawarehouse proxy.

   Execute the following command to create the wproxy database in your DB2 RDBMS:

   ```
   db2 create database wproxy using codeset utf-8 territory US
   ```
2. When creating an Oracle or DB2 database the code set must also be utf-8.

Installing the Warehouse Proxy agent

The Warehouse Proxy is used to upload historical data from agents into the Tivoli Enterprise Data Warehouse for historical reporting. As with any other IBM Tivoli Monitoring 6.1 agent, the Warehouse Proxy agent installation follows the same procedure as the one described in "Installing TEMA on a Windows server" on page 102. Warehouse Proxy agent runs only on the Windows platform.

Database configuration and installation prerequisites

Use the following recommendation when installing RDBMS on DB2 or Oracle.

- DB2

  You must be using at least DB2 V8.2 FP10. You need also to set the environment variable DB2CODEPAGE=1208 as a system environment on the Windows box where the Warehouse Proxy is installed.

  You can find the DB2 fix packs at the following address:

  `http://www.ibm.com/software/data/db2/udb/support/downloadv8.html`

- Oracle

  If you have Oracle 9.2, you must upgrade the ODBC Driver to Version 9.2.0.4. Set the environment variable NLS_LANG=AMERICAN_AMERICA.AL32UTF8 as a system environment on the Windows box where the Warehouse Proxy is installed.

Site for Oracle ODBC drivers:

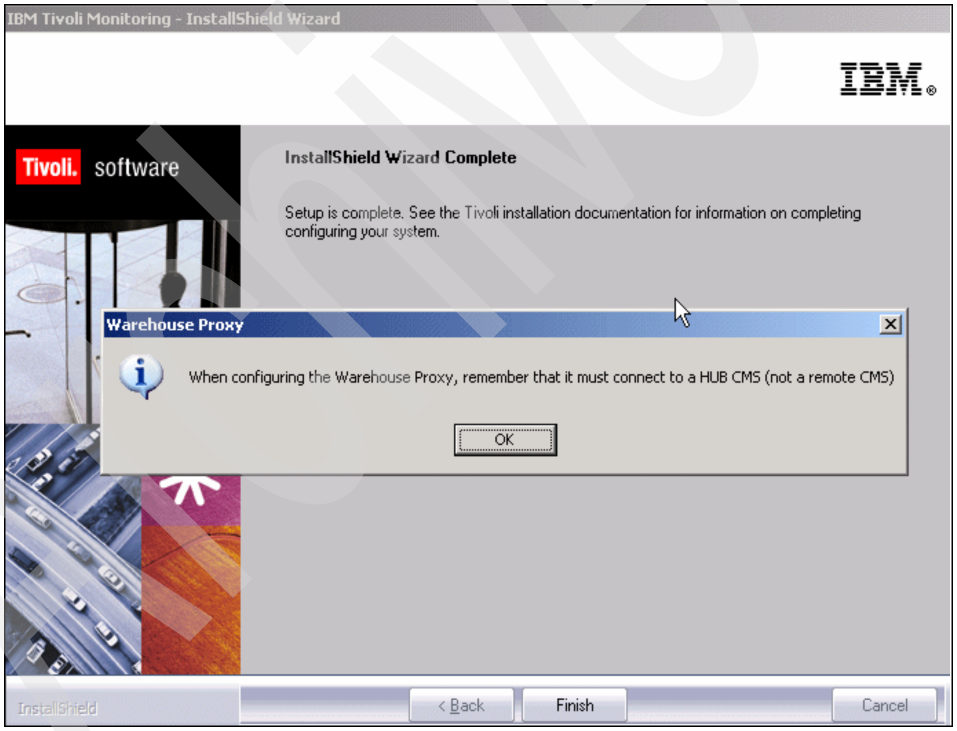
http://www.oracle.com/technology/software/tech/windows/odbc/htdocs/utilsoft.html

Note: IBM plans to provide Warehouse Proxy agent on UNIX and Linux platforms as well, but at GA time only Windows will be supported.

Configuring the Warehouse Proxy agent

The next step after installing the Warehouse Proxy is to configure it to connect to the database in order to insert and retrieve data from the database. The following steps show how to perform the Warehouse Proxy agent configuration.

1. From the Manage Tivoli Enterprise Monitoring Services console, right-click the Warehouse Proxy and select **Reconfigure**.
2. Click **OK** when you see the message shown in Figure 3-43.



InstallShield Wizard Complete

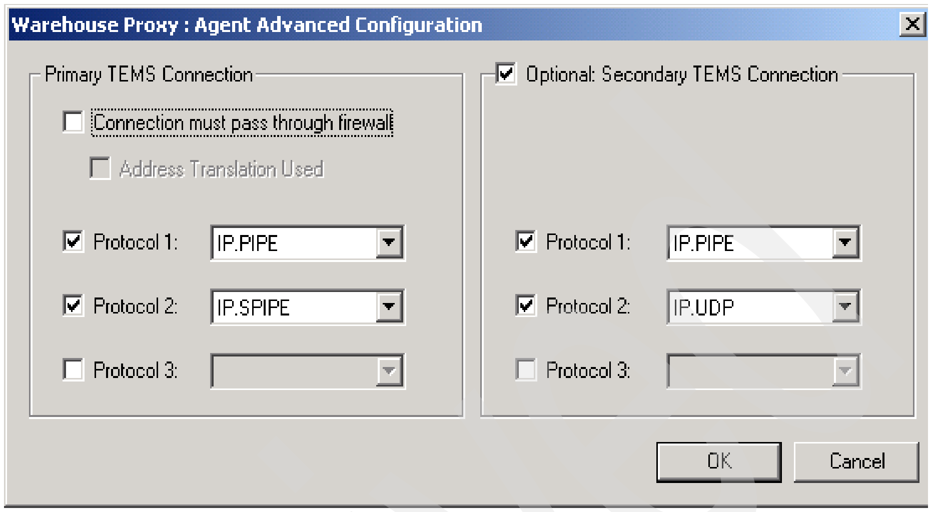
Setup is complete. See the Tivoli installation documentation for information on completing configuring your system.

Warehouse Proxy

When configuring the Warehouse Proxy, remember that it must connect to a HUB CMS (not a remote CMS)

Figure 3-43 Informational window display

3. Configure the protocol communication between the TEMS and the Warehouse Proxy as shown in Figure 3-44 and click **OK**.

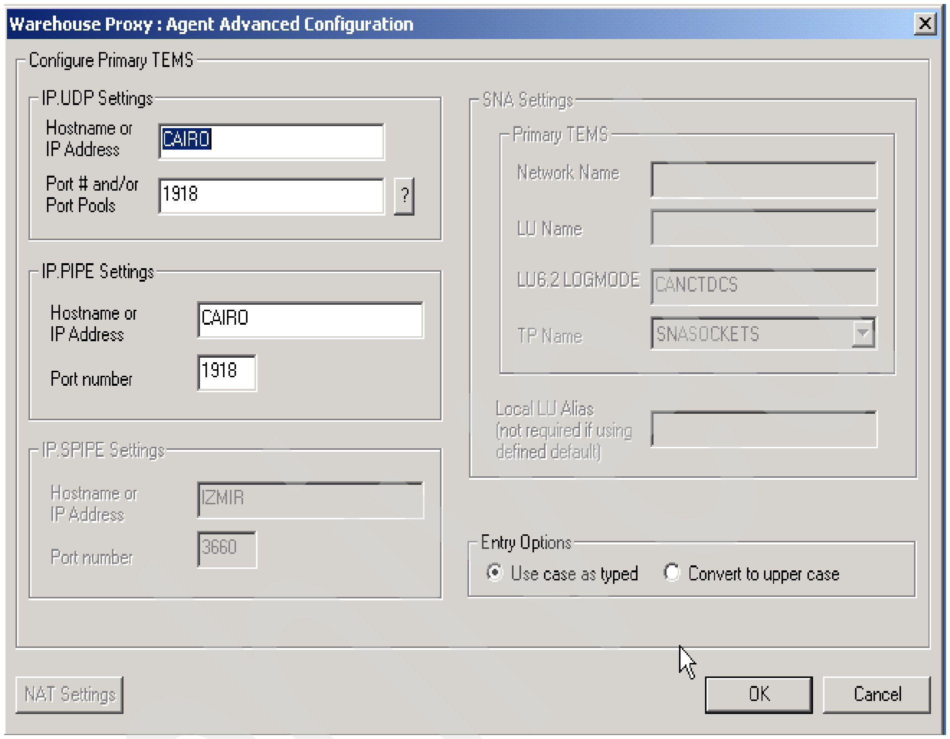

Warehouse Proxy : Agent Advanced Configuration

Primary TEMS Connection
- [ ] Connection must pass through firewall
- [ ] Address Translation Used
- [x] Protocol 1: IP.PIPE
- [x] Protocol 2: IP.SPIPE
- [ ] Protocol 3:

[x] Optional: Secondary TEMS Connection
- [x] Protocol 1: IP.PIPE
- [x] Protocol 2: IP.UDP
- [ ] Protocol 3:

Figure 3-44 Warehouse Proxy agent communication protocol configuration

4. Configure the Hub TEMS host name and the ports where the Warehouse Proxy will connect and click **OK**.


Warehouse Proxy : Agent Advanced Configuration

Configure Primary TEMS

IP.UDP Settings
Hostname or IP Address: CAIRO
Port # and/or Port Pools: 1918

IP.PIPE Settings
Hostname or IP Address: CAIRO
Port number: 1918

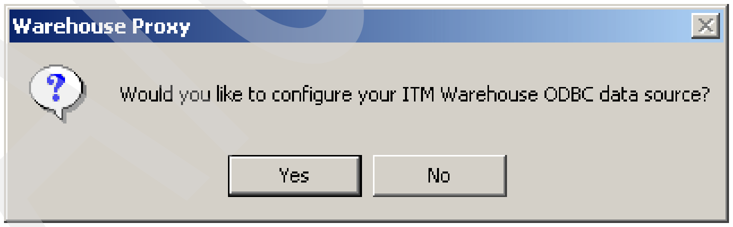
IP.SPIPE Settings
Hostname or IP Address: IZMIR
Port number: 3660

SNA Settings
Primary TEMS
Network Name:
LU Name:
LU6.2 LOGMODE: CANCTDCS
TP Name: SNASOCKETS
Local LU Alias (not required if using defined default):

Entry Options
[✓ Use case as typed] Convert to upper case

Figure 3-45 Warehouse Proxy agent Hub TEMS and port configuration

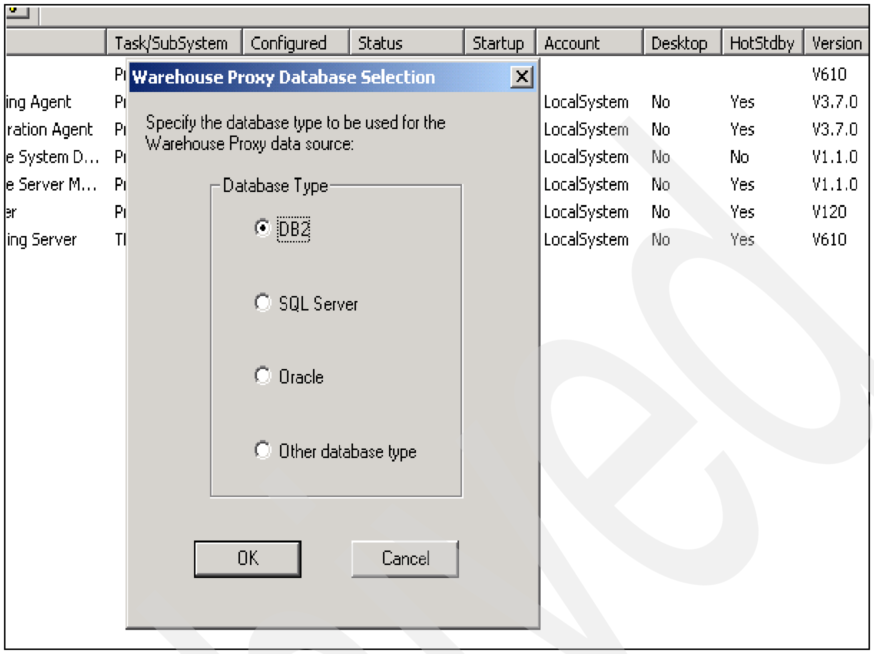
5. Click **OK** when you see the window shown in Figure 3-46.


Warehouse Proxy

Would you like to configure your ITM Warehouse ODBC data source?

Figure 3-46 ITM Warehouse ODBC configuration confirmation window

6. Select the database the Warehouse Proxy agent you will use (Figure 3-47).



	Task/SubSystem	Configured	Status	Startup	Account	Desktop	HotStdby	Version
	Pr							V610
ing Agent	Pr				LocalSystem	No	Yes	V3.7.0
ration Agent	Pr				LocalSystem	No	Yes	V3.7.0
e System D...	Pr				LocalSystem	No	No	V1.1.0
e Server M...	Pr				LocalSystem	No	Yes	V1.1.0
er	Pr				LocalSystem	No	Yes	V120
ing Server	Tl				LocalSystem	No	Yes	V610

Warehouse Proxy Database Selection

Specify the database type to be used for the Warehouse Proxy data source:

Database Type

- [✓ DB2]
- SQL Server
- Oracle
- Other database type

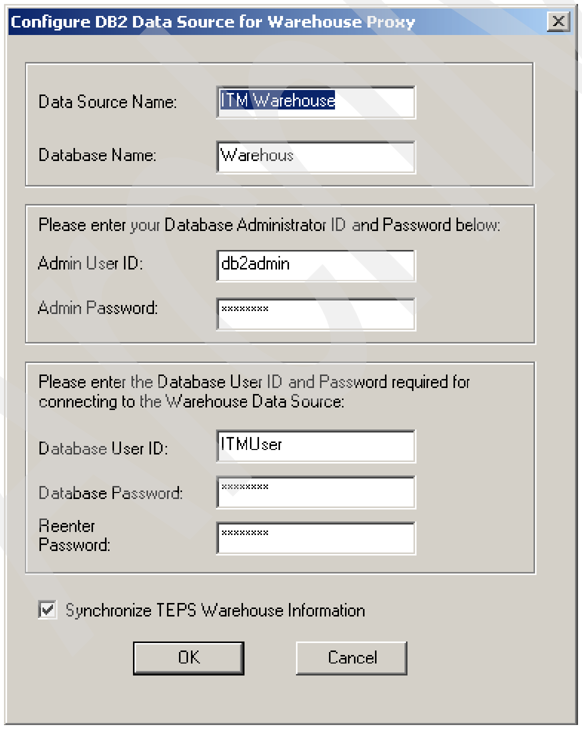
OK Cancel

Figure 3-47 Database selection for Warehouse Proxy configuration

7. Fill in the following fields (Figure 3-50 on page 138) then click **OK**:

Data Source Name	Leave as ITM Warehouse.
Database Name	The name of the database the Warehouse Proxy agent will use to store the data.
Admin User ID	The database user administrator created during database installation (default is db2admin for DB2).
Admin Password	The user database administrator password.
Database User ID	The user ID that will own the table made to store warehouse data. This user must be created on the OS first; refer to "Creating a Windows user" on page 131. The default user is ITMUser.
Database Password	The password of the Database user ID.

Note: After this step completed, the database and the associated tables will be created on your database.


Configure DB2 Data Source for Warehouse Proxy

Data Source Name: ITM Warehouse

Database Name: Warehous

Please enter your Database Administrator ID and Password below:

Admin User ID: db2admin

Admin Password: *********

Please enter the Database User ID and Password required for connecting to the Warehouse Data Source:

Database User ID: ITMUser

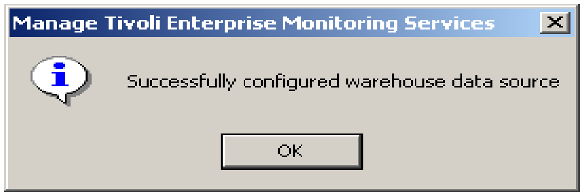
Database Password: *********

Reenter Password: *********

☑ Synchronize TEPS Warehouse Information

Figure 3-48 Data source configuration window for the Warehouse Proxy

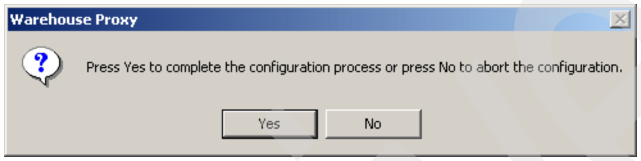
8. Click **OK** on the next pop-up window stating that the Data Warehouse was successfully completed, as shown on Figure 3-49.


Manage Tivoli Enterprise Monitoring Services

Successfully configured warehouse data source

Figure 3-49 Warehouse configuration status message

9. Click **Yes** on the next window (Figure 3-50) to complete the configuration.


Warehouse Proxy

Press Yes to complete the configuration process or press No to abort the configuration.

Figure 3-50 Warehouse Proxy database configuration completion

10. Restart the Warehouse Proxy agent by double-clicking on it.

Important:

You can change the default ODBC data source name using the following procedure:

1. Edit the Windows registry:

```
regedit
```

2. Find this registry key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\CANDLE\KHD\Ver610\Primary\Environment
```

3. Double-click the string **ODBCDATASOURCE** and enter the ODBC data source name of your choice.

3.2.13 Summarization and Pruning agent installation and configuration

Attention: Before you proceed with this section, we strongly recommend reading 4.3, "Planning" on page 209.

The Summarization and Pruning agent is responsible for aggregating historical data and for pruning to the size of the database according to the desired guidelines. This installation is identical to the other agent installation. This section focuses on how to configure your Summarization and Pruning agent.

Installing the Summarization and Pruning agent

Summarization and Pruning agent is similar to any other common agent. To install it, follow the procedure described in "Installing TEMA on a Windows server" on page 102 and use the next section, "Configuring the Summarization and Pruning agent" to make the appropriate configurations.

Configuring the Summarization and Pruning agent

After the Summarization and Pruning agent is installed, configure when and how data will be collected, aggregated, and pruned:

1. From your Windows desktop, click **Start** → **Programs** → **IBM Tivoli Monitoring** → **Manage Tivoli Enterprise Monitoring Services**.

   On a Linux or UNIX system, **cd** to install_dir/bin. Type:

   ```
   ./itmcmd manage
   ```

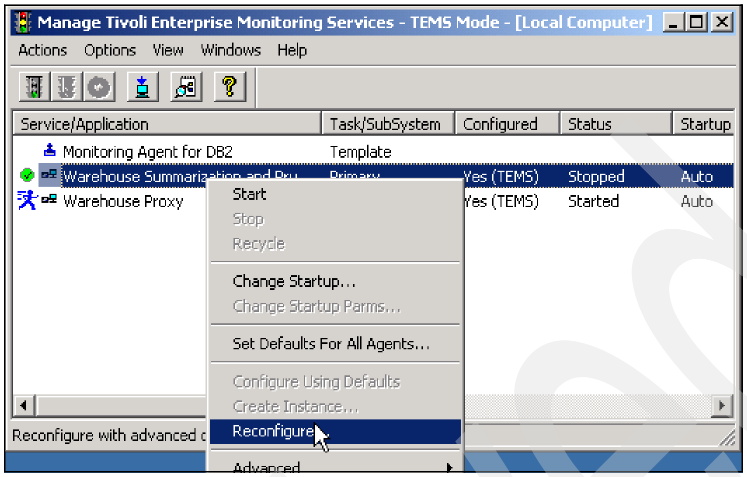
   Note: For more details about this command, execute this phrase:

   ```
   itmcmd manage ?
   ```

   Or see the *IBM Tivoli Monitoring Installation and Setup Guide*, GC32-9407.

2. Right-click **Summarization and Pruning agent**.

3. Click **Reconfigure** as shown in Figure 3-51.

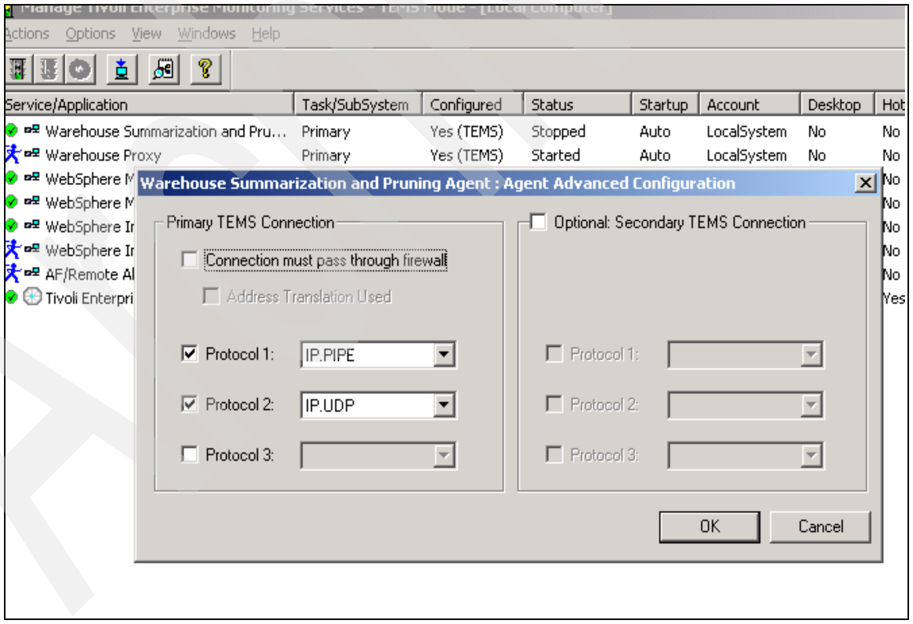

Manage Tivoli Enterprise Monitoring Services - TEMS Mode - [Local Computer]

Service/Application	Task/SubSystem	Configured	Status	Startup
Monitoring Agent for DB2	Template			
Warehouse Summarization and Pru	Primary	Yes (TEMS)	Stopped	Auto
Warehouse Proxy		Yes (TEMS)	Started	Auto

Reconfigure with advanced c

Figure 3-51 Configuring through monitoring console

4. Click **OK** in the Advanced Configuration window (Figure 3-52).


Manage Tivoli Enterprise Monitoring Services - TEMS Mode - [Local Computer]

Service/Application	Task/SubSystem	Configured	Status	Startup	Account	Desktop	Hot
Warehouse Summarization and Pru...	Primary	Yes (TEMS)	Stopped	Auto	LocalSystem	No	No
Warehouse Proxy	Primary	Yes (TEMS)	Started	Auto	LocalSystem	No	No
WebSphere M							No
WebSphere M							No
WebSphere Ir							No
WebSphere Ir							No
AF/Remote Al							No
Tivoli Enterpri							Yes

Warehouse Summarization and Pruning Agent : Agent Advanced Configuration

Primary TEMS Connection
- [ ] Connection must pass through firewall
- [ ] Address Translation Used
- [x] Protocol 1: IP.PIPE
- [x] Protocol 2: IP.UDP
- [ ] Protocol 3:

- [ ] Optional: Secondary TEMS Connection
- [ ] Protocol 1:
- [ ] Protocol 2:
- [ ] Protocol 3:

Figure 3-52 Configuring Summarization and Pruning agent connection protocol

5. Click **OK** in the next window.

6. Click **Yes** in the Warehouse Summarization and Pruning agent window to configure the Summarization and Pruning agent.
7. On the Sources tab, enter the Tivoli Data Warehouse database and Tivoli Enterprise Portal server information. Before performing any update, confirm that the default configuration is accurate. If it is not, use the following procedure to update the information:
   a. In the JDBC™ drivers field:
      i. Click **Add** to invoke the file browser window to select your JDBC driver. The default is:

         `DB2: C:\Program Files\IBM\SQLLIB\java\db2java.zip`

         Click **OK** to close the browser and add the JDBC drivers to the list.
      ii. To delete a driver, highlight its entry in the JDBC drivers list and click **Delete**.

      This gives you the ability to collect JDBC drivers to communicate with your Tivoli Data Warehouse database. JDBC drivers are installed separately and each database provides a set of these JDBC drivers.

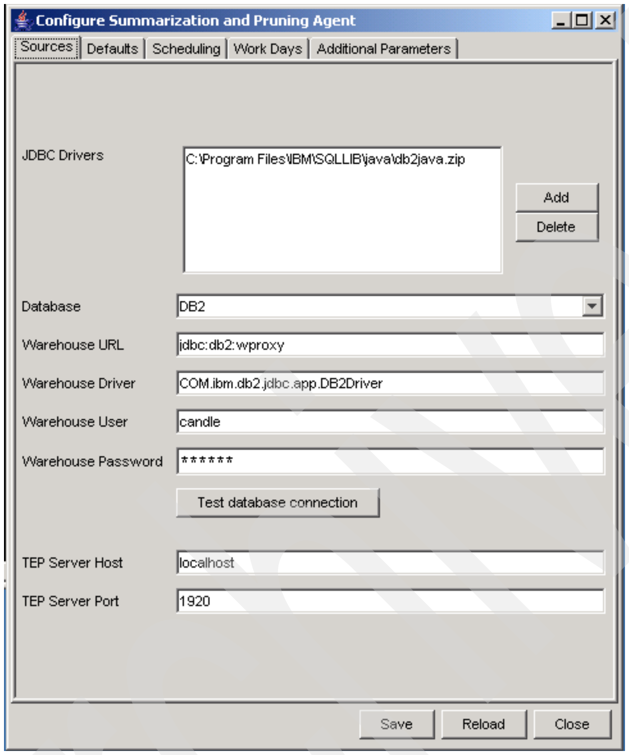
      Notes:
      - If your Tivoli Data Warehouse database is on UNIX, find the directory where your database is installed and in the jdbc drivers directory, select only the db2jcc.jar and db2java.zip files. For example: <db2installdir>/java/db2jcc.jar and db2java.zip.
      - If your Tivoli Data Warehouse database is on MS SQL Server, install the JDBC drivers from the Microsoft SQL Server Web site. These are the three files that you need:
        – c:\Program Files\Microsoft SQL Server 2000 Driver for JDBC\lib\msbase.jar
        – c:\Program Files\Microsoft SQL Server 2000 Driver for JDBC\lib\mssqlserver.jar
        – c:\Program Files\Microsoft SQL Server 2000 Driver for JDBC\lib\msutil.jarv. In the pull-down list, select the type of database for your Tivoli Data Warehouse.

   b. Enter the Warehouse URL, Driver, Schema, user ID and password.

   Important: During the configuration of the Warehouse Proxy, a database user (called ITMUser by default) is created. The user ID that you enter here must match that database user.

c. Click **Test database connection** to ensure you can communicate with your Tivoli Data Warehouse database.

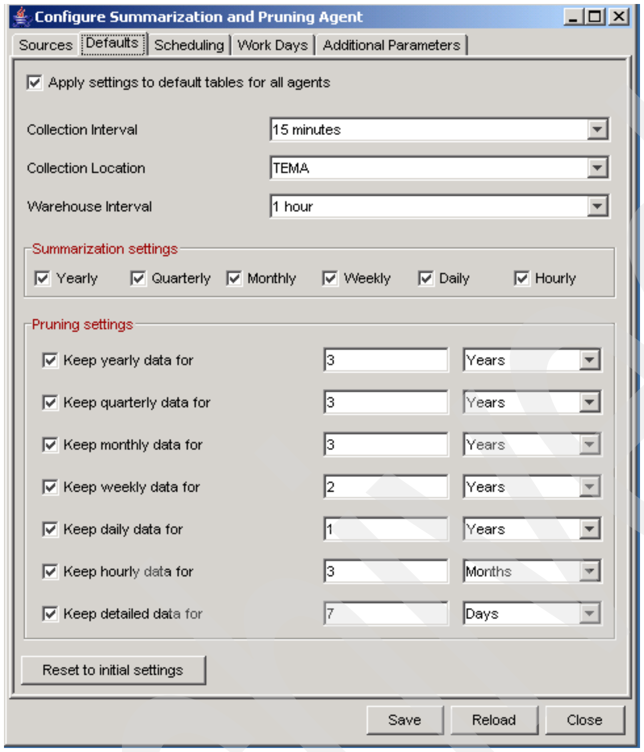
d. Enter the Tivoli Enterprise Portal Server host and port if you do not want to use the defaults.



JDBC Drivers	C:\Program Files\IBM\SQLLIB\java\db2java.zip
Database	DB2
Warehouse URL	jdbc:db2:wproxy
Warehouse Driver	COM.ibm.db2.jdbc.app.DB2Driver
Warehouse User	candle
Warehouse Password	******
TEP Server Host	localhost
TEP Server Port	1920

Figure 3-53 Configuring agent TEPS and database connection

8. Click the **Defaults** tab and select the settings for your summarization and pruning information (Figure 3-53 on page 142).


Configure Summarization and Pruning Agent

Sources | Defaults | Scheduling | Work Days | Additional Parameters

☑ Apply settings to default tables for all agents

Collection Interval	15 minutes
Collection Location	TEMA
Warehouse Interval	1 hour

Summarization settings

☑ Yearly ☑ Quarterly ☑ Monthly ☑ Weekly ☑ Daily ☑ Hourly

Pruning settings

☑ Keep yearly data for	3	Years
☑ Keep quarterly data for	3	Years
☑ Keep monthly data for	3	Years
☑ Keep weekly data for	2	Years
☑ Keep daily data for	1	Years
☑ Keep hourly data for	3	Months
☑ Keep detailed data for	7	Days

Reset to initial settings

Save Reload Close

Figure 3-54 Configuring how data will be collected and pruned

– Clicking **Reset** returns all settings in this window to the default settings.

– If you do not want to use the defaults, select the appropriate time periods you want in the Summarization section to change your summarization values.

– To change your Pruning settings:

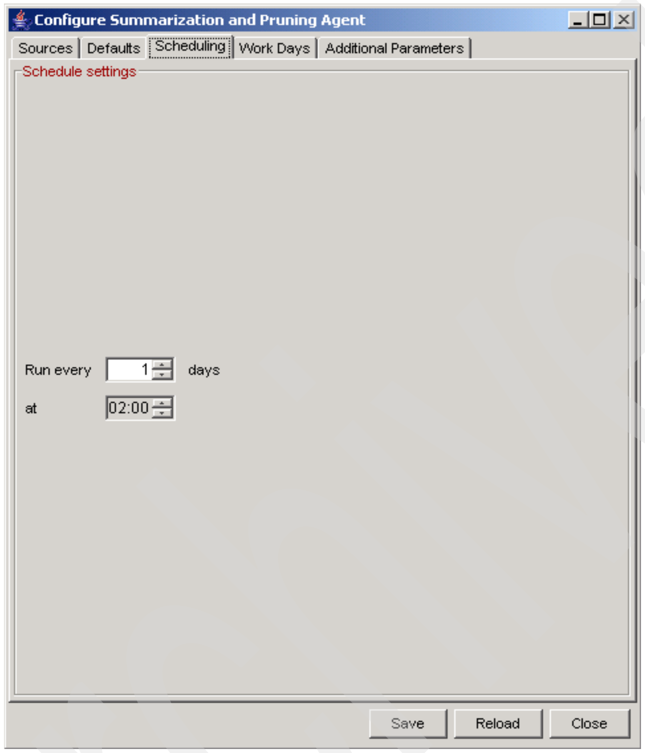
  i. Select the time periods for the pruning of your data: **Keep yearly data for**, **Keep quarterly data for**, and so on.

  ii. Enter the number of time periods you wish in the next field.

  iii. Select the time period you wish. For example, if you want to prune hourly data when it becomes 30 days old, select Hourly, keep 30 and choose Days as the time period from the drop down list.

– Select **Apply settings to default tables for all agents** to keep the changes you have made.

9. Click the **Scheduling** tab and select the scheduling information (Figure 3-55).
   - Schedule the agent to run every x days.
   - Select the hour of the day that you want the summarization to run. The default is to run every day at 2 a.m.



Run every 1 days

at 02:00

Figure 3-55 Scheduling the data collection and pruning

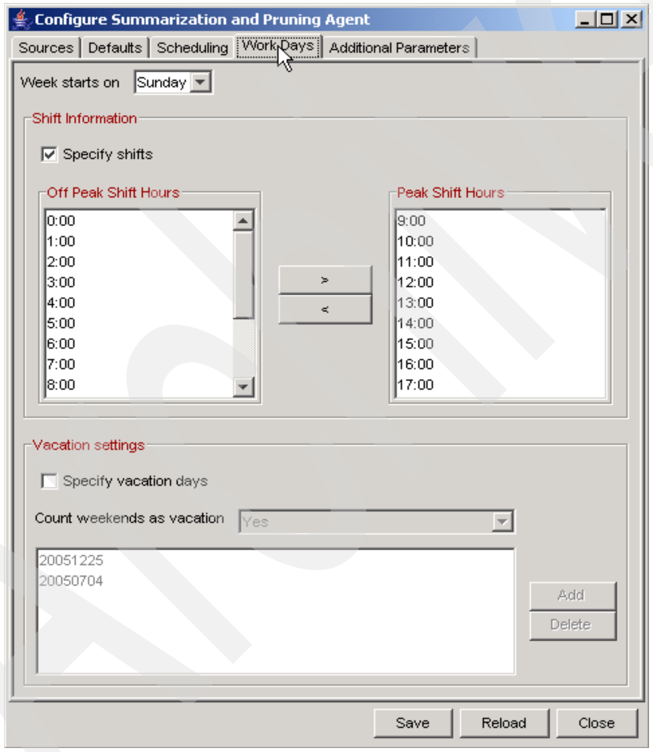
10. Click the **Work Days** tab (Figure 3-56 on page 145) and specify shift information and vacation settings:
    - Select day the week starts on.
    - If you want to specify shifts, select **Specify shifts**. The default settings for this field are listed in the Peak Shift Hours box on the right side of the window. You can change these settings by selecting the hours you want in the Off Peak Shift Hours box and clicking the right arrow button to add them to the Peak Shift Hours box.

Note: Changing the shift information after data has been summarized can create an inconsistency in the data. Data that has been collected and summarized cannot be recalculated with the new shift values.

- To change your vacation settings, select **Specify vacation days**. If you do not want to set your vacation days, do not select this check box.
  1. Click **Yes** or **No** to count weekends as vacation days.
  2. Click **Add** to add vacation days, and select the vacation days you want to add from the calendar.

     Note: On UNIX or Linux, right-click, instead of left-click, to select the month and year.

  3. The days you select appear in the box below the Select vacation days field. If you want to delete any days you have previously chosen, select them and click **Delete** (Figure 3-56**)**.


Configure Summarization and Pruning Agent

Sources | Defaults | Scheduling | Work Days | Additional Parameters

Week starts on: Sunday

Shift Information
- [✓] Specify shifts

Off Peak Shift Hours: 0:00, 1:00, 2:00, 3:00, 4:00, 5:00, 6:00, 7:00, 8:00

Peak Shift Hours: 9:00, 10:00, 11:00, 12:00, 13:00, 14:00, 15:00, 16:00, 17:00

Vacation settings
- [ ] Specify vacation days

Count weekends as vacation: Yes

20051225
20050704

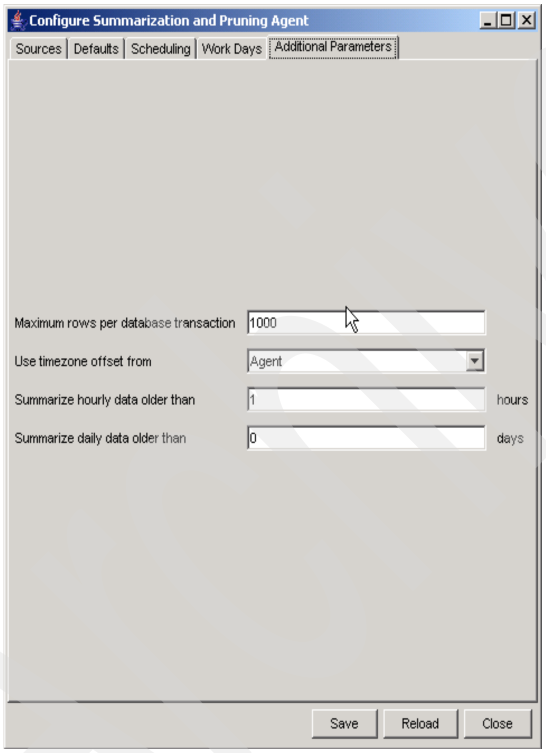
Figure 3-56 Defining shift periods and vacation settings

11. Click the **Additional Parameters** tab (Figure 3-57):

   a. Specify the maximum number of rows that can be deleted in a single database transaction. The values are 1 through n. The default is 1000.

   b. Specify the age of the hourly and daily data you want summarized. Values are 0 through n. The default is 1 for hourly data and 0 for daily data.

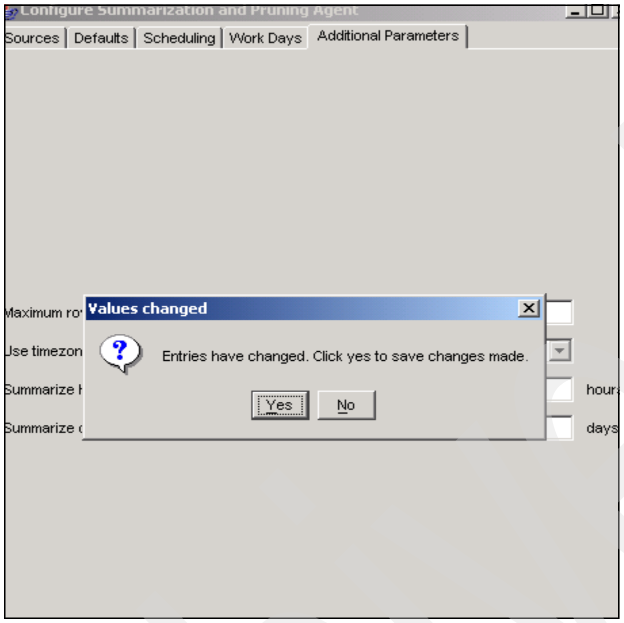
   c. Choose the time zone you want to use from the pull-down list. If the Tivoli Data Warehouse and agents that are collecting data are all not in the same time zone, and all the data is stored in the same database, use this option to identify the time zone you want to use.



Maximum rows per database transaction 1000
Use timezone offset from Agent
Summarize hourly data older than 1 hours
Summarize daily data older than 0 days

Figure 3-57 Configuring additional parameters

12. Click **Yes** on the next window to save your configuration (Figure 3-58).



Entries have changed. Click yes to save changes made.

Figure 3-58   Saving the Pruning and Summarization agent configuration

12. The following buttons are visible only on a UNIX or Linux system:
    - Click **Save** after you have all your settings correct.
    - Click **Reload** to reload the original values.
    - Click **Cancel**, at any time, to cancel out of the ConfigureSummarization and Pruning agent window. You are prompted to save any data you have changed.

Changing configuration settings using the History Collection Configuration window in the Tivoli Enterprise Portal

To change the default data summarization, pruning configurations, or both after installing the Summarization and Pruning agent, use the History Collection Configuration window in the Tivoli Enterprise Portal. Refer to 4.4.2, "History configuration" on page 231.

3.2.14 Event synchronization installation

Event synchronization components enable IBM Tivoli Monitoring 6.1 to send events to IBM Tivoli Enterprise Console server, as well as IBM Tivoli Monitoring 6.1 users to view events from IBM Tivoli Enterprise Console in the Tivoli

Enterprise Portal. Install the IBM Tivoli Enterprise Console on the event server and configure the TEMS to send events to the IBM Tivoli Enterprise Console server to have those features available. Table 3-8 provides an overview of the steps required to install and configure the IBM Tivoli Enterprise Console.

Table 3-8 TEC event synchronization installation and configuration steps

Step	Reference
1. Gather information required during the installation and configuration processes.	"Information to gather before you begin" on page 148
2. Install the IBM Tivoli Enterprise Console on your event server.	"Installing event synchronization on your event server" on page 149
3. Configure your monitoring server to forward events to IBM Tivoli Enterprise Console.	"Configuring the monitoring server to forward events" on page 160
4. Start and stop IBM Tivoli Enterprise Console on the monitoring server.	"Starting and stopping the process that sends updates to a monitoring server" on page 161

Information to gather before you begin

You need the following information to successfully install and configure event synchronization between IBM Tivoli Monitoring and IBM Tivoli Enterprise Console:

- Host names (or IP addresses), user IDs, and passwords for the monitoring servers that you want to receive events from.
- Simple Object Access Protocol (SOAP) information to send events to a monitoring server (the URL, the rate to send requests to the server).
- Event rule base information (either the name of a new rule base to create or the name of an existing rule base to use).

Notes:

- If your IBM Tivoli Enterprise Console event server is running on Windows 2003 and you are planning to install the IBM Tivoli Enterprise Console remotely (using a program such as Terminal Services to connect to that Windows 2003 computer), run the **`change user /install`** command before you run the installation. This puts the computer into the required "install" mode. After the installation, run the **`change user /execute`** command to return the computer to its previous mode.
- On a UNIX computer, you must configure your TCP/IP network services in the /etc/hosts file to return the fully qualified host name.

Installing event synchronization on your event server

You can install the event synchronization either through an installation wizard or from the command line.

Note: The IBM Tivoli Enterprise Console event server must be recycled during this installation process.

Configuring SOAP on the Hub TEMS

Simple Object Access Protocol (SOAP) is a communications XML-based protocol that enables applications to exchange information through the Internet. SOAP is platform independent and language independent. SOAP uses XML to specify a request and reply structure. It uses HTTP as the transport mechanism to drive the request and to receive a reply.

By default, all monitoring servers are enabled for Web Services. Use the following sections to configure IBM Tivoli Monitoring Web Services (SOAP server) on Windows XP Professional Edition or Windows 2000 computers.

The instructions in this section assume that you have a basic understanding of SOAP, XML and XML Namespaces, and the Web Services Description Language (WSDL).

For complete information about customizing the SOAP interface for your site, refer to *IBM Tivoli Monitoring Administrator's Guide,* SC32-9408.

Table 3-9 outlines the steps required to configure SOAP.

Table 3-9 SOAP configuration steps

Steps	References
1. Define the hubs with which your SOAP server communicates.	"Defining the Hub TEMS that communicate with SOAP" on page 149
2. Create users and grant them access.	"Adding users" on page 150
3. Verify that you have successfully configured SOAP.	"Verifying the SOAP configuration" on page 151

Defining the Hub TEMS that communicate with SOAP

In this step you use Manage Tivoli Monitoring Services to activate the SOAP server and define hubs with which the SOAP server communicates.

To define the SOAP hubs:

1. Open **Manage Tivoli Monitoring Services**.

2. Right-click **Tivoli Enterprise Monitoring Server**.
3. Click **Advanced** → **Configure SOAP Server Hubs**.
4. Click **Add Hub**. The Hub Specification window opens.
5. Select the communications protocol from the Protocol menu.
6. Specify an alias name in the Alias field (for example: SOAP).
7. If you are using TCP/IP or TCP/IP Pipe communications, complete the following fields:
   - Hostname/ip address: Corresponds to the host name or IP address of your Hub TEMS.
   - Port: Default is 1920 (as the one to connect to TEPS).

   Note: If you are connecting to a remote monitoring server, the protocol information must be identical to that used for the hub monitoring server.

8. Click **OK**.

Adding users

In this step you define users on each Hub and specify the access rights for each user (query or update).

Use the following steps:

1. Select the server (click anywhere within the server tree displayed), if necessary.
2. Under Add User Data, type the user name. User IDs must be identical to those specified for monitoring server logon validation. Access is restricted to only that monitoring server to which a user has access.

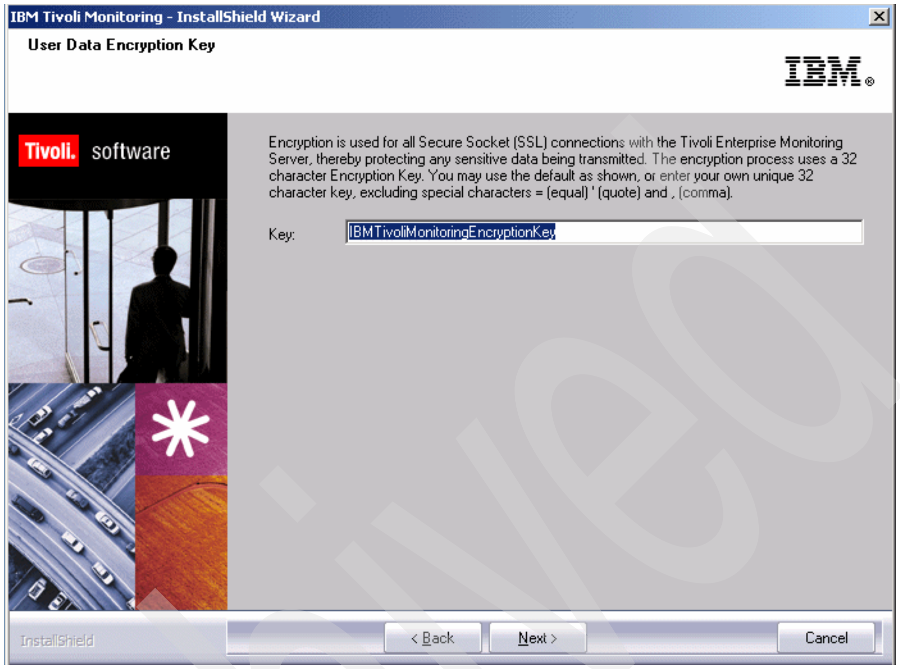
   Attention: If you do not supply a user ID, all users are given permission to update data.

3. Click the type of user access: **Query** or **Update**.
4. Click **Add User**. The server tree is updated, showing the user and type of access.

   To delete a user, select the user name from the tree and click **Delete Item**.

   To delete a hub, click anywhere in the hub’s tree and click **Clear Tree**.

When done you should see a window similar to Figure 3-59.


IBM Tivoli Monitoring - InstallShield Wizard

User Data Encryption Key

Encryption is used for all Secure Socket (SSL) connections with the Tivoli Enterprise Monitoring Server, thereby protecting any sensitive data being transmitted. The encryption process uses a 32 character Encryption Key. You may use the default as shown, or enter your own unique 32 character key, excluding special characters = (equal) ' (quote) and , (comma).

Key: IBMTivoliMonitoringEncryptionKey

Figure 3-59 SOAP server hub configuration

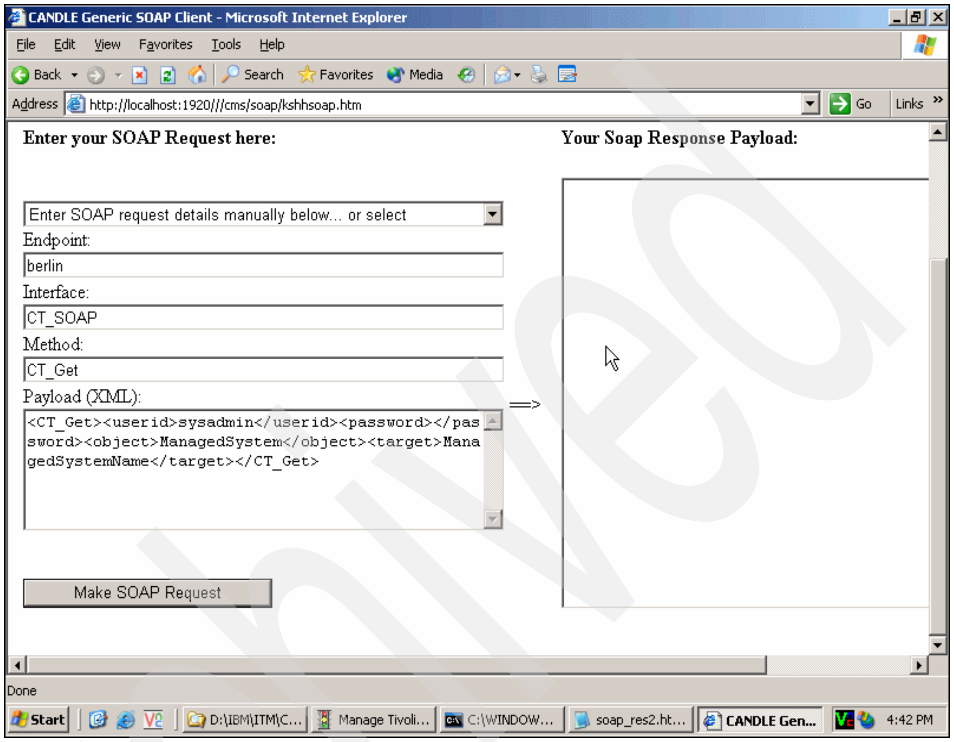
Verifying the SOAP configuration

In this step you verify that SOAP has been configured properly by starting the SOAP client and making a request using the Internet Explorer Web browser:

1. From your Hub TEMS type the following address in your Internet browser and press Enter.

```
http://localhost:1920///cms/soap/kshhsoap.htm
```

2. In the window that opens, enter the type of SOAP request and the endpoint where you want to retrieve the data (Figure 3-60). Click **Make SOAP Request**.


CANDLE Generic SOAP Client - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost:1920///cms/soap/kshhsoap.htm
Go Links
Enter your SOAP Request here:
Your Soap Response Payload:
Enter SOAP request details manually below... or select
Make SOAP Request
Done
Start
CANDLE Gen...
4:42 PM

Endpoint: berlin

Interface: CT_SOAP

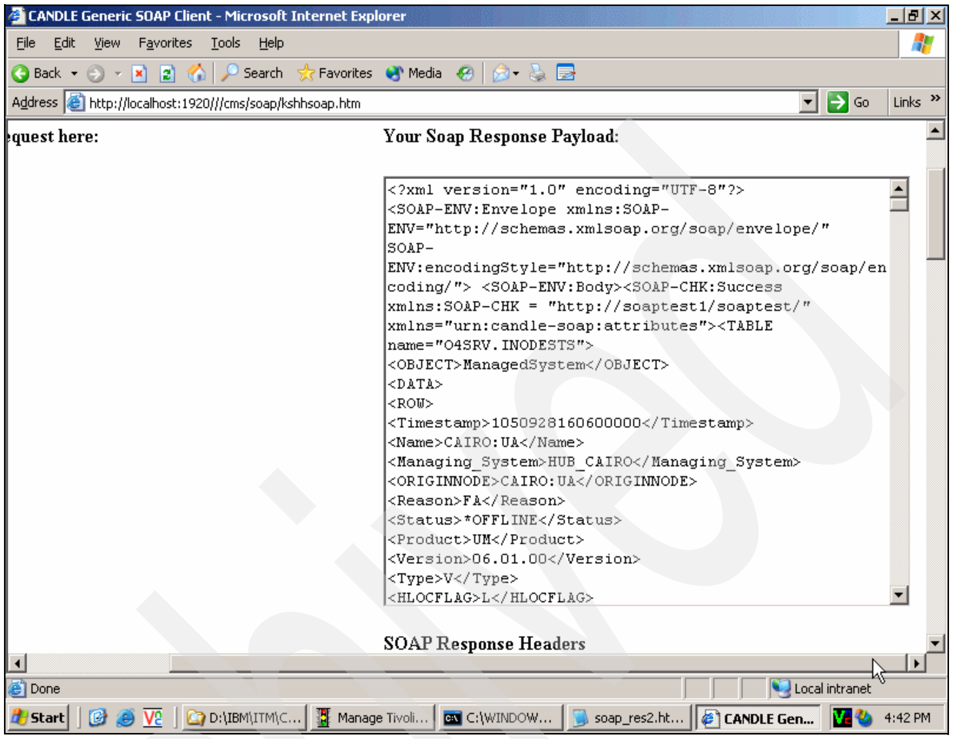
Method: CT_Get

Payload (XML):

```
<CT_Get><userid>sysadmin</userid><password></password><object>ManagedSystem</object><target>ManagedSystemName</target></CT_Get>
```

Figure 3-60 SOAP Web interface configuration test

3. This displays the page shown in Figure 3-61.



CANDLE Generic SOAP Client - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost:1920///cms/soap/kshhsoap.htm
Go Links
equest here:
Your Soap Response Payload:
SOAP Response Headers
Done
Local intranet
Start D:\IBM\ITM\C... Manage Tivoli... C:\WINDOW... soap_res2.ht... CANDLE Gen... 4:42 PM

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/en
coding/"> <SOAP-ENV:Body><SOAP-CHK:Success
xmlns:SOAP-CHK = "http://soaptest1/soaptest/"
xmlns="urn:candle-soap:attributes"><TABLE
name="O4SRV.INODESTS">
<OBJECT>ManagedSystem</OBJECT>
<DATA>
<ROW>
<Timestamp>1050928160600000</Timestamp>
<Name>CAIRO:UA</Name>
<Managing_System>HUB_CAIRO</Managing_System>
<ORIGINNODE>CAIRO:UA</ORIGINNODE>
<Reason>FA</Reason>
<Status>*OFFLINE</Status>
<Product>UM</Product>
<Version>06.01.00</Version>
<Type>V</Type>
<HLOCFLAG>L</HLOCFLAG>
```

Figure 3-61 SOAP Response

Installing event synchronization from a wizard

Use the following steps to install event synchronization from the installation wizard.

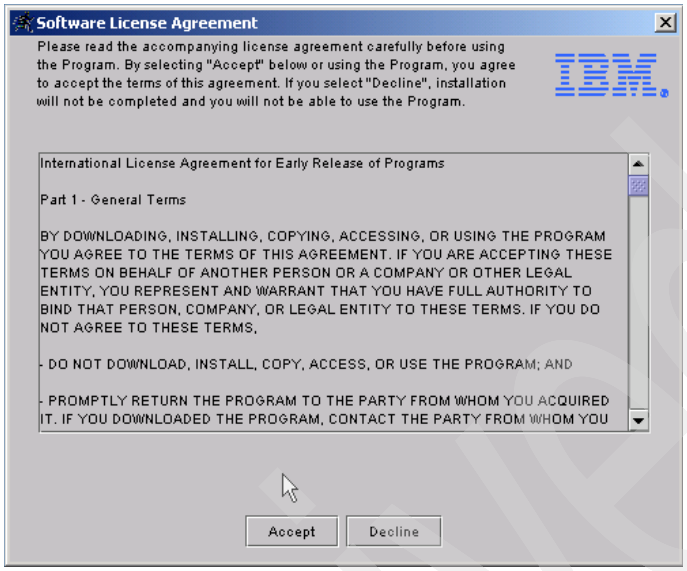
> **Note:** If your IBM Tivoli Enterprise Console Server is a UNIX machine, execute the following procedure using a local GUI or an X-Terminal.

1. On the event server, launch the IBM Tivoli Enterprise Console installation by executing the following command:

   ```
   cd $install_dir/unix/tec/
   ./setupAIX.bin
   ```

2. Click **Next** on the Welcome window.

3. Select **Accept** in the license agreement window (Figure 3-62).



Please read the accompanying license agreement carefully before using the Program. By selecting "Accept" below or using the Program, you agree to accept the terms of this agreement. If you select "Decline", installation will not be completed and you will not be able to use the Program.

International License Agreement for Early Release of Programs

Part 1 - General Terms

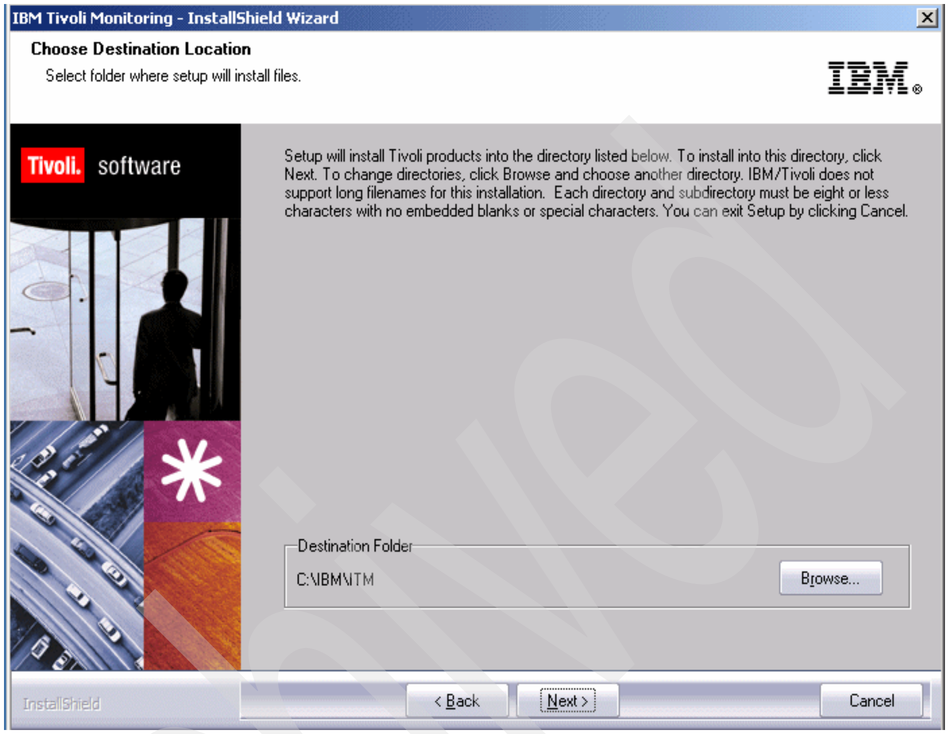
BY DOWNLOADING, INSTALLING, COPYING, ACCESSING, OR USING THE PROGRAM YOU AGREE TO THE TERMS OF THIS AGREEMENT. IF YOU ARE ACCEPTING THESE TERMS ON BEHALF OF ANOTHER PERSON OR A COMPANY OR OTHER LEGAL ENTITY, YOU REPRESENT AND WARRANT THAT YOU HAVE FULL AUTHORITY TO BIND THAT PERSON, COMPANY, OR LEGAL ENTITY TO THESE TERMS. IF YOU DO NOT AGREE TO THESE TERMS,

- DO NOT DOWNLOAD, INSTALL, COPY, ACCESS, OR USE THE PROGRAM; AND

- PROMPTLY RETURN THE PROGRAM TO THE PARTY FROM WHOM YOU ACQUIRED IT. IF YOU DOWNLOADED THE PROGRAM, CONTACT THE PARTY FROM WHOM YOU

Figure 3-62 Event synchronization Software License Agreement window

4. Click **Next**.


IBM Tivoli Monitoring - InstallShield Wizard

Choose Destination Location

Select folder where setup will install files.

Setup will install Tivoli products into the directory listed below. To install into this directory, click Next. To change directories, click Browse and choose another directory. IBM/Tivoli does not support long filenames for this installation. Each directory and subdirectory must be eight or less characters with no embedded blanks or special characters. You can exit Setup by clicking Cancel.

Destination Folder

C:\IBM\ITM

Figure 3-63 Event synchronization configuration fields

5. Complete the fields and click **Next**.

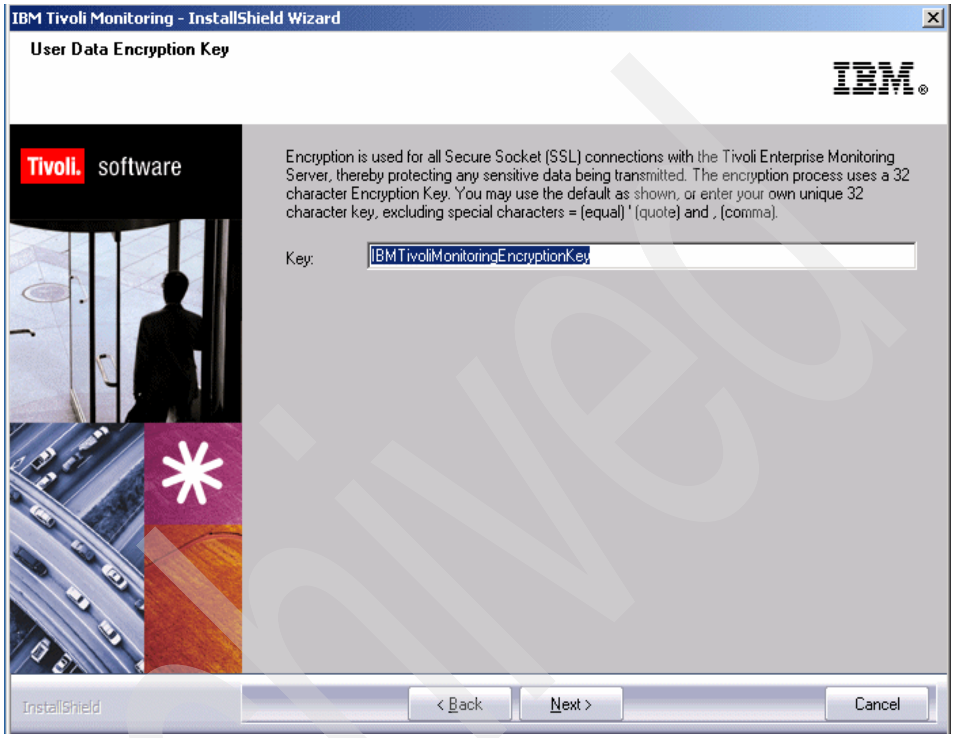
   Table 3-10 shows each configuration field and its respective description.

Table 3-10 Tivoli Enterprise Console event synchronization configuration fields

Fields	**Description**
Name of the configuration file	The name of the file where event synchronization configuration information is stored. The default name is situpdate.conf.
Number of seconds to sleep when no situation updates	The polling interval, in seconds. The minimum (and default) value is 1. If there are no situation events, the process that forwards events to IBM Tivoli Enterprise Console will rest for 1 second.

Fields	Description
Number of bytes to use to save last events	Number of bytes that the long-running process will use when it saves the location of the last event it processes. This value must be an integer. The minimum (and default) is 50.
URL of the monitoring server SOAP server	The URL for the SOAP server configured on the computer where the monitoring server is running. The default value is cms/soap. This value is used to create the URL to which IBM Tivoli Enterprise Console sends event information. For example, `http://hostname:port///cms/soap`, where `hostname` is the host name of the monitoring server and `port` is the port used by that server.
Rate for sending events from IBM Tivoli Enterprise Console to TEMS via Web services	The maximum number of events sent to the monitoring server at one time. The minimum (and default) value is 10 events.
Level of debug detail for log	The level of information for event synchronization that will be logged. You have the following options: ► Low (default) ► Medium ► Verbose

6. Complete the information about the files where events will be written as described in Table 3-11 when the window in Figure 3-64 pops up, and click **Next**.


IBM Tivoli Monitoring - InstallShield Wizard

User Data Encryption Key

Encryption is used for all Secure Socket (SSL) connections with the Tivoli Enterprise Monitoring Server, thereby protecting any sensitive data being transmitted. The encryption process uses a 32 character Encryption Key. You may use the default as shown, or enter your own unique 32 character key, excluding special characters = (equal) ' (quote) and , (comma).

Key: IBMTivoliMonitoringEncryptionKey

Figure 3-64 Event synchronization cache file configuration window

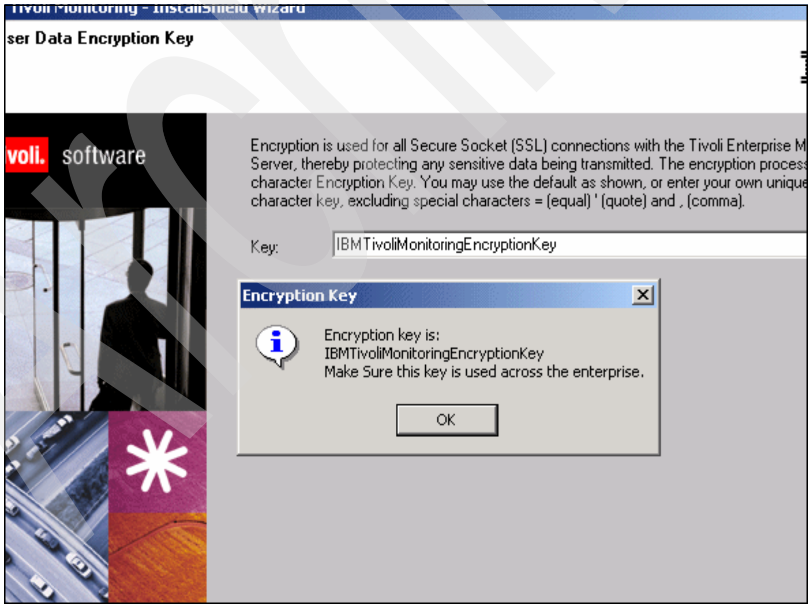
Table 3-11 TEC event synchronization caches file config fields description

Fields	Description
Maximum size of any single cache file, in bytes	The maximum permitted size, in bytes, for any one event cache file. The minimum (and default) value is 50000. Do not use commas (as in 50,000) when specifying this value.
Maximum number of cache files	The maximum number of event caches files at any given time. The minimum (and default) value is 10. When this value is reached, the oldest file is deleted to make room for a new file.

Fields	Description
Directory for cache files to reside	The location where event cache files are located. The default locations are as follows: ► On Windows: C:\tmp\TME\TEC\OM_TEC\persistence ► On UNIX: /var/TME/TEC/OM_TEC/persistence

7. Provide the host name, user ID, and password for each monitoring server with which you want to synchronize events and click **Add**. You must specify information for at least one monitoring server. Figure 3-65 shows an example of TEMS information during the event synchronization configuration.

Host name	The fully qualified host name for the computer where the monitoring server is running. This should match the information that will be in events coming from this monitoring server.
User ID	The user ID to access the computer where the monitoring server is running.
Password	The password to access the computer.
Confirmation	The password confirmation.



Encryption is used for all Secure Socket (SSL) connections with the Tivoli Enterprise M
Server, thereby protecting any sensitive data being transmitted. The encryption process
character Encryption Key. You may use the default as shown, or enter your own unique
character key, excluding special characters = (equal) ' (quote) and , (comma).

Key: IBMTivoliMonitoringEncryptionKey

Encryption Key

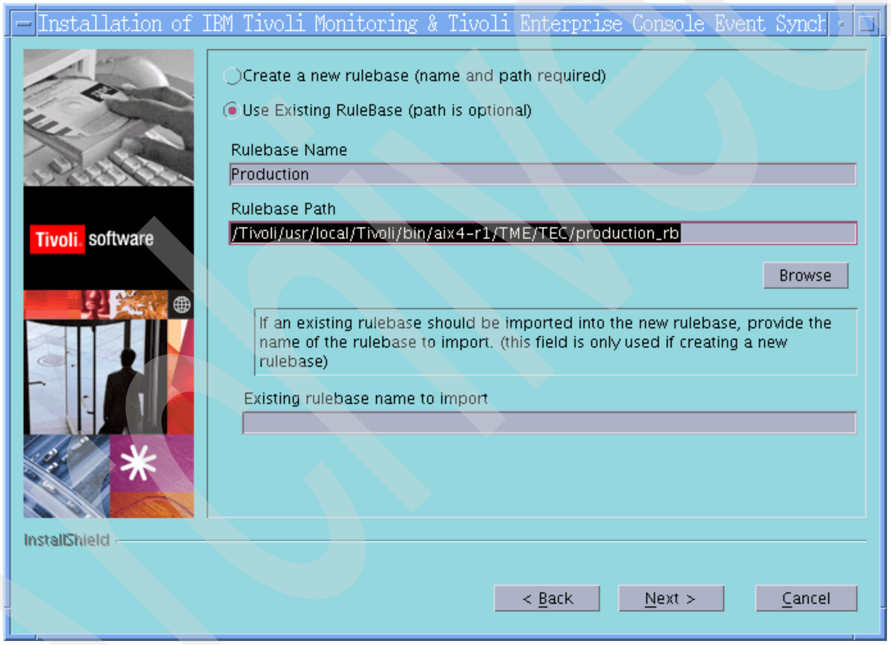
Encryption key is:
IBMTivoliMonitoringEncryptionKey
Make Sure this key is used across the enterprise.

OK

Figure 3-65 Event synchronization Tivoli Enterprise Monitoring Server information

8. When you have provided information about all of the monitoring servers, click **Next**.
9. Specify the rule base that you want to use to synchronize events (Figure 3-66). You can either:
   – Create a new rule base.
   – Use an existing rule base. If you select to use an existing rulebase, the IBM Tivoli Enterprise Console BAROC class files (omegamon.baroc and Sentry.baroc [if not present]) and the omegamon.rls ruleset file are imported into your existing rulebase.

   We used the second option as we already have IBM Tivoli Enterprise Console set up and running properly.


Installation of IBM Tivoli Monitoring & Tivoli Enterprise Console Event Synch
Tivoli software
InstallShield
Browse
< Back
Next >
Cancel

( ) Create a new rulebase (name and path required)
(●) Use Existing RuleBase (path is optional)

Rulebase Name
Production

Rulebase Path
/Tivoli/usr/local/Tivoli/bin/aix4-r1/TME/TEC/production_rb

If an existing rulebase should be imported into the new rulebase, provide the name of the rulebase to import. (this field is only used if creating a new rulebase)

Existing rulebase name to import

Figure 3-66 IBM Tivoli Enterprise Console rule base configuration

Notes:

- If you are creating a new rule base, type the name for the rule base you want to create and the path to the new rule base location. You must specify a location as there is no default.
- If you are using an existing rule base, type the name of the rule base.
- If you want to import an existing rule base into a new rule base, type the name of the existing rule base in the Existing rulebase to import field. This step is available only if you are creating a new rule base.

10. Click **Next**.
11. Click **Next** on the preinstallation summary panel. The installation begins.
12. When the installation and configuration steps have completed, click **Finish** on the Summary Information window.

Note: If any configuration errors occurred during installation and configuration, you are directed to a log file that contains additional troubleshooting information.

If you did not configure the event forwarding during the Hub TEMS installation and configuration, refer to "Configuring the monitoring server to forward events" on page 160.

Configuring the monitoring server to forward events

Before the monitoring server forwards any situation events to IBM Tivoli Enterprise Console, you have to enable that forwarding and the filtering of events.

Enabling event forwarding

Use the following steps to enable event forwarding on your monitoring server:

1. From your Hub TEMS, in Manage Tivoli Monitoring Services, right-click the monitoring server and click **Reconfigure** (Windows) or **Configure** (UNIX).
2. On the configuration options window, select **TEC Event Integration Facility**. For UNIX, click the **TEC** tab to view this configuration option.
3. Click **OK** and then **OK** again.

4. Complete the following fields on the IBM Tivoli Enterprise Console Server: Location and Port Number window and click **OK**:

   TEC Server Location

   The host name or IP address (in dotted format, such as 0.0.0.0) for the computer where the IBM Tivoli Enterprise Console event server is installed.

   TEC Port Number

   The port number for the event server. Set this value to 0 (when your event server is a UNIX server) unless the portmapper is not available on the event server (as when the event server is running on a Windows server). When you specify 0, the port number is retrieved by the portmapper.

   If you want to use the default port (5529), edit the tec_installdir/TME/TEC/.tec_config file on the event server and remove the semicolon (;) from the following line: tec_recv_agent_port=5529. Port 5529 is often used when the event server is a Windows platform.

After you have done this, any error during the installation can be found in the error log file, /tmp/tec_sync_install.log.

5. Click **Finish.**

Enabling event filtering

By default, all situation events are filtered out, meaning that they are never forwarded to IBM Tivoli Enterprise Console. Use the following steps to change the filtering on the TEMS servers that forward events to IBM Tivoli Enterprise Console:

1. Open the om_tec.config file (located in <install_dir>\cms\TECLIB\ on Windows and <install_dir>/tables/<tems_name>/TECLIB on UNIX).
2. Comment out the following line by adding a pound sign (#) in front of it:

   ```
   Filter:Class=ITM_Generic;master_reset_flag='';
   ```

3. Save and close the file.

Starting and stopping the process that sends updates to a monitoring server

To send event updates to a monitoring server, you must start a long-running process called Situation Update Forwarder. This process is started automatically when the event server starts. To stop the process manually, change to the

$BINDIR/TME/TEC/OM_TEC/bin directory (where $BINDIR is the location of the IBM Tivoli Enterprise Console installation) and run the following command:

On Windows:

```
stopSUF.cmd
```

On UNIX:

```
stopSUF.sh
```

On Windows, you can also use the Tivoli Situation Update Forwarder service to start or stop the forwarding of event updates. You can start and stop this service either from the Windows Service Manager utility or with the following commands:

```
net start situpdate
net stop situpdate
```

To start the process, run the following command:

On Windows:

```
startSUF.exe config_file
```

On UNIX:

```
startSUF.sh config_file
```

Note: config_file is the name of the file where IBM Tivoli Enterprise Console configuration information is stored. The default name is /etc/TME/TEC/OM_TEC/situpdate.conf. Refer to Figure 3-63 on page 155 and Table 3-10 on page 155 for more details.

3.2.15 Configuring the Hot Standby

The optional Hot Standby function enables you to maintain continuous availability by defining a standby monitoring server to provide backup for your hub monitoring server. If the hub monitoring server fails, hub functions automatically switch to the standby monitoring server. IBM Tivoli Monitoring automatically connects all remote monitoring servers and agents to the standby monitoring server. There is no automatic switch that returns control to the hub monitoring server when it is available. If you want to switch back to the hub monitoring server, you must manually stop the standby monitoring server.

Configuring Hot Standby involves the following steps:

1. Install the monitoring server software on the systems you want to use as Hot Standby (refer to “Installing the backup hub monitoring server” on page 163). We assume that the primary Hub TEMS is already up and running.

2. Configure Hot Standby on the hub monitoring server, the backup monitoring server, and any remote monitoring servers associated with the hub monitoring server; refer to “Configuring Hot Standby” on page 163.
3. Configure Hot Standby on any agents that are associated with the hub monitoring server like the TEPS and the Warehouse Proxy agent; refer to “Configuring the Warehouse Proxy” on page 165.

Note: TEMS Hot Standby is not supported on z/OS. For an alternative setup on z/OS systems, refer to 12.3.3, “z/OS Moveable HUB” on page 681.

Installing the backup hub monitoring server

See “Installing and configuring a Hub TEMS on a Windows server” on page 14 or “Installing a Hub TEMS on a UNIX server” on page 116 for information about installing a hub monitoring server. When you are installing the backup hub monitoring server, use identical values to those you used when installing the primary hub monitoring server.

Configuring Hot Standby

Configure the hub server, the standby hub server, and any remote servers associated with the hub server.

Note: The hub and standby monitoring servers should be configured as mirrors of each other.

Configuring Hot Standby on a Windows TEMS

Use the following procedure to configure Hot Standby on a Windows TEMS:

1. In Manage Tivoli Monitoring Services, right-click the name of the hub monitoring server and click **Reconfigure** (or **Configure** on UNIX).
2. Select **Configure Standby CMS** and specify the protocols used by the standby server. These protocols should match those specified for the hub server.
3. Click **OK**, then **OK** again on the message that is displayed.
4. Click **OK** on the window that displays the communication settings for this server.
5. Type the host name or IP address for the standby monitoring server in the Hostname or IP Address field and click **OK**.
6. Configure the Tivoli Enterprise Console server name and port in the next window. The default will be the one you configure on your Hub TEMS.
7. Restart the monitoring server.

8. Repeat these steps for the standby monitoring server and any remote monitoring servers.

Configuring Hot Standby on a UNIX TEMS

Use the following procedure to configure Hot Standby on a UNIX TEMS:

1. From the $CANDLEHOME/bin execute the following command:

   ```
   ./itmcmd config -S -t tems_name
   ```

   `tems_name` is the name of the TEMS hosted by the UNIX server. In Example 3-14, server MADRID is being configured with server cairo as its Hot Standby server.

Example 3-14 ./itmcmd config -S -t HUB_MADRID output

```
[root@madrid][/opt/IBM/ITM/bin]-> ./itmcmd config -S -t HUB_MADRID
CandleConfig       :  installer level  400 / 100.
CandleConfig       :  running aix516 jre.
Configuring CMS...
```

2. Press Enter if *LOCAL is the default. If not, type `*LOCAL`.

   ```
   Hub or Remote [*LOCAL or *REMOTE] (Default is: *LOCAL): *LOCAL
   TEMS Host Name (Default is: madrid):
   ```

3. Select your primary communication protocol.

   ```
   Network Protocol 1 [ip, sna, or ip.pipe] (Default is: ip.pipe):
   ```

4. Select a secondary communication protocol.

   ```
     Now choose the next protocol from one of these:
     - ip
     - sna
     - none
   Network Protocol 2 (Default is: ip):
   ```

5. Select a third communication protocol.

   ```
     Now choose the next protocol from one of these:
     - sna
     - none
   Network Protocol 3 (Default is: none):
   ```

6. Choose the port used by the server being configured.

   ```
   IP Port Number (Default is: 1918):
   IP.PIPE Port Number (Default is: 1918):
   ```

7. Accept the default for the next option or enter the correct file name.

   ```
   Enter name of KDC_PARTITION (Default is: null):
   Enter path and name of KDC_PARTITIONFILE (Default is:
   /opt/IBM/ITM/tables/HUB_MADIRD/partition.txt):
   ```

8. Press Enter to accept the default.

```
Configuration Auditing? [YES or NO] (Default is: YES):
Now enter with the server host name used as hot standby (backup)
Hot Standby TEMS Host Name (Default is: milan):
```

9. Enter the communication protocol and the port (Example 3-15).

Example 3-15 Entering communication protocol and the port

```
Hot Standby Protocol 1 [ip, sna,ip.pipe or ip.spipe] (Default is: ip.pipe):

     Now choose the next protocol from one of these:
     - ip
- sna
     - ip.spipe
     - none
Hot Standby Protocol 2 (Default is: ip):

     Now choose the next protocol from one of these:
     - sna
- ip.spipe
     - none
Hot Standby Protocol 3 (Default is: none):
Hot Standby IP Port Number (Default is: 1918):
Hot Standby IP.PIPE Port Number (Default is: 1918):
```

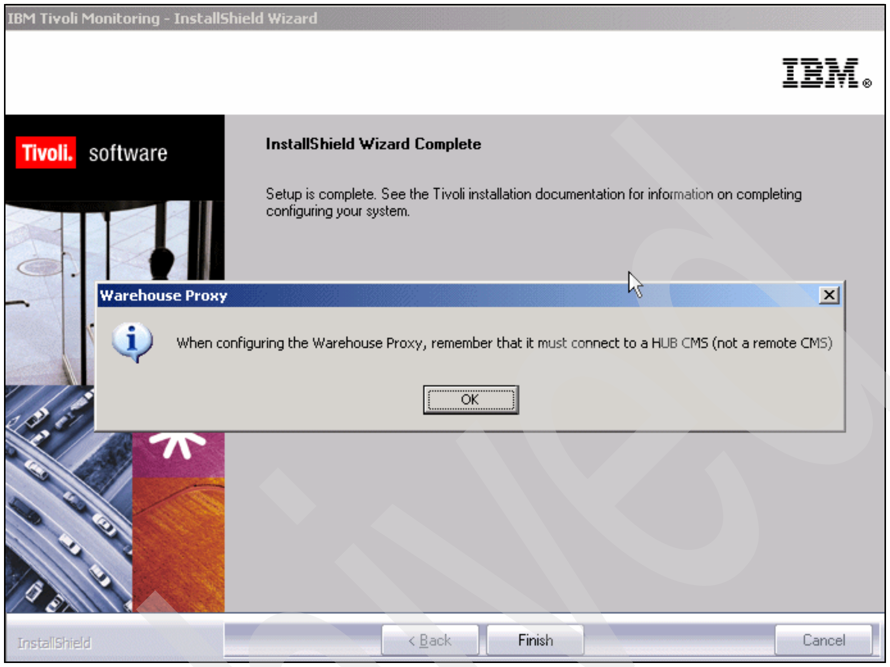
The next steps are similar to those in “Configuring the UNIX monitoring server” on page 172.

Configuring the Warehouse Proxy

The Warehouse Proxy must be configured to point to a Secondary TEMS in case the primary Hub TEMS fails and the Hot Standby takes place. Use the following procedure to configure the Warehouse Proxy:

1. Open the Manage Tivoli Monitoring Services console from the Warehouse proxy server, right-click on the **Warehouse Proxy** server, and click **Reconfigure**.

2. Click **OK** on the pop-up window.



IBM Tivoli Monitoring - InstallShield Wizard

InstallShield Wizard Complete

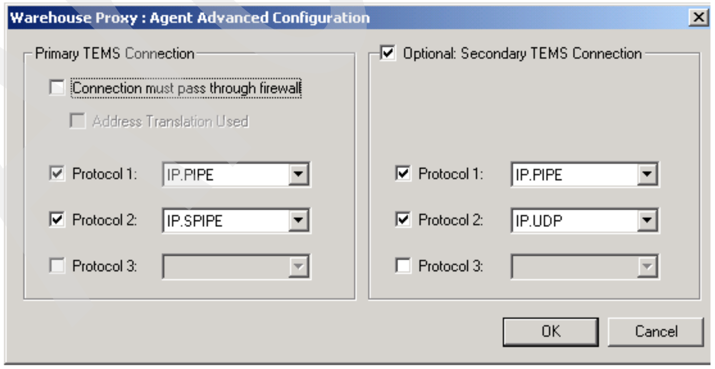
Setup is complete. See the Tivoli installation documentation for information on completing configuring your system.

Warehouse Proxy

When configuring the Warehouse Proxy, remember that it must connect to a HUB CMS (not a remote CMS)

Figure 3-67 Warehouse Proxy confirmation window configuration

3. The primary TEMS communication protocol is already defined. Check the **Optional Secondary TEMS Connection** check box and configure the communication protocol the secondary TEMS will be using as shown in Figure 3-68.



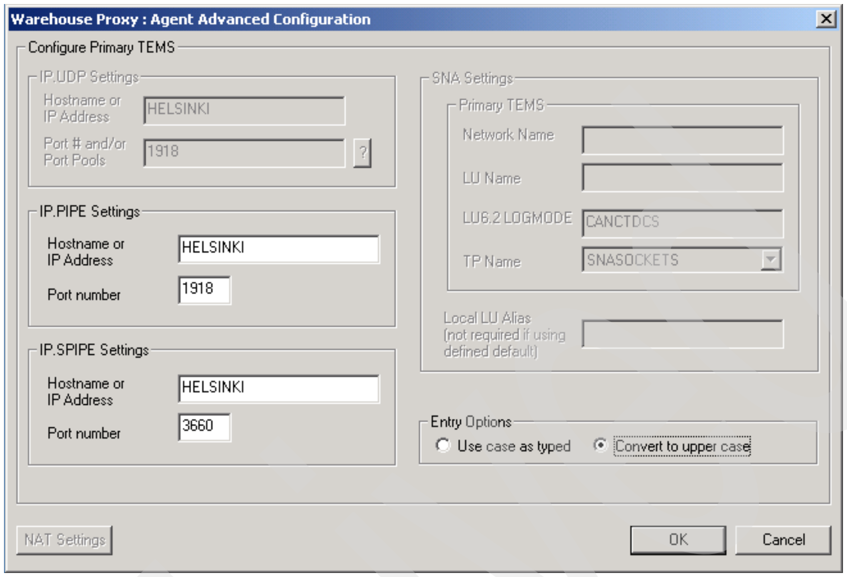
Warehouse Proxy : Agent Advanced Configuration

Primary TEMS Connection
- [ ] Connection must pass through firewall
- [ ] Address Translation Used
- [✓] Protocol 1: IP.PIPE
- [✓] Protocol 2: IP.SPIPE
- [ ] Protocol 3:

[✓] Optional: Secondary TEMS Connection
- [✓] Protocol 1: IP.PIPE
- [✓] Protocol 2: IP.UDP
- [ ] Protocol 3:

Figure 3-68 Warehouse Proxy Secondary TEMS communication configuration

4. Select **OK** on the next window or update the primary TEMS if it has changed (Figure 3-69).


Warehouse Proxy : Agent Advanced Configuration

Configure Primary TEMS

IP.UDP Settings
Hostname or IP Address: HELSINKI
Port # and/or Port Pools: 1918

IP.PIPE Settings
Hostname or IP Address: HELSINKI
Port number: 1918

IP.SPIPE Settings
Hostname or IP Address: HELSINKI
Port number: 3660

SNA Settings
Primary TEMS
Network Name:
LU Name:
LU6.2 LOGMODE: CANCTDCS
TP Name: SNASOCKETS
Local LU Alias (not required if using defined default):

Entry Options
Use case as typed [✓ Convert to upper case]

Figure 3-69 Warehouse Proxy primary TEMS configuration

5. Enter the secondary TEMS host name for each communication protocol chosen on Figure 3-68 on page 166 and click **OK.**
6. The Warehouse Proxy is thus configured to connect to the secondary Hub TEMS if the primary one fails.

Configuring the TEPS

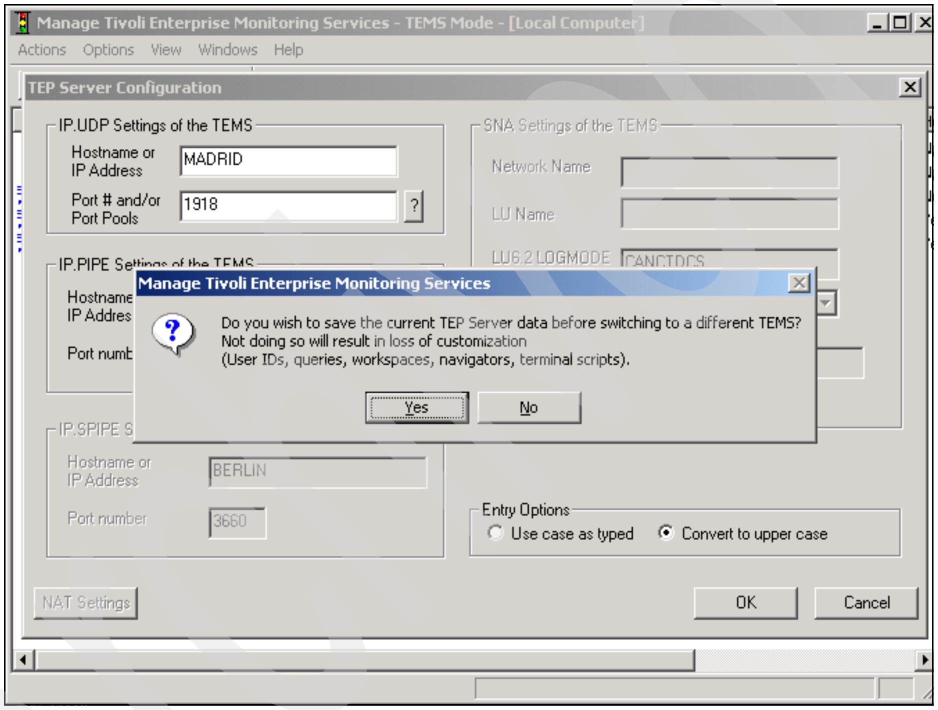
The TEPS will have to be configured only if the primary Hub TEMS fails and the standby Hub TEMS becomes the primary. Use the following steps to configure the TEPS:

1. Open the Manage Tivoli Monitoring Services console.
2. Right-click the **Tivoli Enterprise Portal Server service**.
3. Select **Reconfigure** (Window) or **Configure** (Linux).
4. Specify the new Hub TEMS and click **OK.**
5. The window shown in Figure 3-70 on page 168 appears. Click **OK**.

Note: Clicking OK as shown in Figure 3-70 on page 168 saves the user IDs, queries, workspace, navigators, and terminal scripts in %CANDLEHOME%\CNPS\CMS\TEMS_HOSTNAME_PORT\saveexport.sql.

(*TEMS_HOSTNAME* is the new TEMS host name and *PORT* is the port number the TEPS will connect to the TEMS. For example:

```
F:\IBM\ITM\CNPS\CMS\MADRID_1918\saveexeport.sql
```



Do you wish to save the current TEP Server data before switching to a different TEMS?
Not doing so will result in loss of customization
(User IDs, queries, workspaces, navigators, terminal scripts).

Entry Options: ( ) Use case as typed [✓ Convert to upper case]

Figure 3-70 TEPS configuration window database backup confirmation

6. From the Enterprise Monitoring Server console, restart the TEPS.

Note: You do not have to reconfigure the TEPS database.

3.2.16 Installing and configuring the scenario 2 environment

This section describes the installation and configuration of IBM Tivoli Monitoring 6.1 components on an UNIX Hub TEMS environment.

Installing a Hub TEMS on a UNIX server

To install a Hub TEMS on a UNIX server, you can use either root or a different user with the proper permissions on the IBM Tivoli Monitoring 6.1 files and directories; see "Creating an IBM Tivoli account on UNIX servers" on page 65.

Use the following steps to install the monitoring server on a UNIX computer.

Notes:

- We will not show the installation of the Hub TEMS individually; we will only focus on how to install a Hub TEMS on UNIX environment.
- This procedure is also valid for Linux servers, so we will not make any distinction between UNIX and Linux configurations in this section.

1. In the directory where you extracted the installation files, run this command:

   ```
   ./install.sh
   ```

2. When prompted for the IBM Tivoli Monitoring home directory, press Enter to accept the default (opt/IBM/ITM). If you want to use a different installation directory, type the full path to that directory and press Enter.

   Note: If the directory you specified does not exist, you are asked whether to create it. Type `y` to create this directory.

   This displays the prompt shown in Example 3-16.

Example 3-16 Select one of the following prompt

```
Select one of the following:
1) Install products to the local host.
2) Install products to depot for remote deployment (requires TEMS).
3) Exit install.
```

Type **1** to start the installation process and press Enter. The following menu will be displayed to you (only the relevant information are shown to you):

Example 3-17 Software Licensing Agreement

```
Software Licensing Agreement
1. Czech
2. English
```

```
3. French
4. German
5. Italian
6. Polish
7. Portuguese
8. Spanish
9. Turkish
Please enter the number that corresponds to the language
you prefer.
```

3. Type the number that corresponds to the language that you want to display the software license agreement in and press Enter.
4. Press Enter to display the license agreement.
5. Type `1` to accept the agreement and press Enter. The message in Example 3-18 appears.

Example 3-18 Preparing to install the Global Security Kit message

```
runGSkit        :  Preparing to install the Global Security Kit.
runGSkit warning:  the 'root' ID or password is required for this phase,
continuing ...
 Will enable automatic agent initiation after reboot.

Please enter root password or press Enter twice to skip.
```

Notes:

- If you are installing the IBM Tivoli Monitoring 6.1 with root user you will not obviously be asked the root password for the GSKit installation.
- GSKit is Global Security Kit, an IBM toolkit that enables products to provide secure connections over TCP/IP among IBM Tivoli Monitoring 6.1 components.

  Two main functions to GSKit:

  – A toolkit that can be used for Secure Socket Layer (SSL) communications using public key encryption/decryption methodology
  – Key management functions using iKeyman (a Java-based application for managing keys and key requests)

6. Type the root password and press Enter if you want to use GSKit; otherwise press Enter twice.
7. Type a 32-character encryption key and press Enter. If you want to use the default key, press Enter without typing any characters.

Important: Save this key to use when you install the other components.

8. Type the number for the operating system for which you want to install products when the following menu is displayed, then press Enter.

Example 3-19 List of available OSs for IBM Tivoli Monitoring 6.1 installation

```
Product packages are available in /opt/itm61s1/unix

Product packages are available for the following operating systems and
component support categories:

 1) AIX R5.1 (32 bit)
 2) AIX R5.1 (64 bit)
 3) AIX R5.2 (32 bit)
 4) AIX R5.2 (64 bit)
 5) AIX R5.3 (32 bit)
 6) AIX R5.3 (64 bit)
 7) Solaris R10 (32 bit)
 8) Solaris R10 (64 bit)
 9) Solaris R8 (32 bit)
10) Solaris R8 (64 bit)
11) Solaris R9 (32 bit)
12) Solaris R9 (64 bit)

Type the number for the OS or component support category you want, or type "q"
to quit selection
[ number "5" or "AIX R5.3 (32 bit)" is default ]:  5
```

Note: The default value is your current operating system.

9. Type `y` to confirm the operating system and press Enter. A numbered list of available components is displayed.
10. Type the number that corresponds to the Tivoli Enterprise Monitoring Server option and press Enter.
11. The available products to be installed are listed. You can choose more than one product by typing their corresponding numbers separated by a comma or a space. Example 3-20 shows the option list.

Example 3-20 Option list

```
The following products are available for installation:

 1) Monitoring Agent for UNIX Logs  V06.10.00.00
 2) Monitoring Agent for UNIX OS  V06.10.00.00
 3)Summarization and Pruning agent  V06.10.00.00
 4) Tivoli Enterprise Monitoring Server  V06.10.00.00
 5) Tivoli Enterprise Services User Interface  V06.10.00.00
 6) Universal Agent  V06.10.00.00
 7) all of the above

Type the numbers for the products you want to install, or type "q" to quit
selection.
If you enter more than one number, separate the numbers by a comma or a space.

Type your selections here:  4
```

12. Type `y` to confirm your selection or `n` to restart.
13. When prompted, type a name for your monitoring server:

    ```
    Please enter TEMS name: HUB_MADRID
    ```

    Do not use the fully qualified host name. Press Enter.
14. After all of the components are installed, you are asked whether you want to install components for a different operating system. Type `n` and press Enter.

Installation is complete. The next step is to configure your monitoring server.

Configuring the UNIX monitoring server

Use the following steps to configure the hub monitoring server.

> **Note:** When we accept the default value on the various displayed options, we just press Enter. Thus, for some cases in the following examples you will not see any entry, because we simply press Enter.

1. At the command line, change to the opt/IBM/ITM/bin directory (or the directory where you installed IBM Tivoli Monitoring, actually the $CANDLEHOME/bin).

   ```
   cd $CANDLEHOME/bin
   ```
2. Run the following command:

   ```
   ./itmcmd config -S -t tems_name
   ```

   This produces the message shown in Example 3-21.

Example 3-21 itmcmd config output

```
CandleConfig        :  installer level  400 / 100.
CandleConfig        :  running li6243 jre.
Configuring CMS...

Hub or Remote [*LOCAL or *REMOTE] (Default is: *LOCAL):*LOCAL
TEMS Host Name (Default is: madrid): (ENTER (hitted)
```

   Choose *LOCAL (the default) as you are installing a Hub TEMS.

3. Configure the communication protocol. Several options are offered: IP, SNA, IP.PIPE,IP.SPIPE. Select your preferred protocol. Refer to “Communications protocol selection” on page 37 for detailed information about protocol specifications.
4. If you want to use a secondary protocol in case the first one fails, enter it at the next option window (Example 3-22).

Example 3-22 Entering a secondary protocol

```
   Now choose the next protocol from one of these:
        - ip
        - sna
        - ip.spipe
        - none
   Network Protocol 2 (Default is: ip):

        Now choose the next protocol from one of these:
        - sna
        - ip.spipe
        - none
   Network Protocol 3 (Default is: none):

Network Protocol 1 [ip, sna, ip.pipe or ip.spipe] (Default is: ip.pipe):
   Now choose the next protocol from one of these:
        - ip
        - sna
        - ip.spipe
        - none
   Network Protocol 2 (Default is: ip):
   Now choose the next protocol from one of these:
        - sna
        - ip.spipe
        - none
Network Protocol 3 (Default is: none):
```

   We have chosen IP.PIPE as the primary protocol and IP.UDP as the secondary protocol.

5. Set the ports for the communication protocols you have chosen; the default is 1918. You can choose another port if you want to, but you have to remember to use it when setting up the Remote TEMS and the TEMA that will be connected to the Hub TEMS. We select the default configuration:

   ```
   IP Port Number (Default is: 1918):
   IP.PIPE Port Number (Default is: 1918):
   ```

6. You are asked for the name of KDC_PARTITION as shown in Example 3-23 on page 174. This is used in environments using NAT across a firewall. We select the default.

Example 3-23 KDC_PARTITION question

```
Enter name of KDC_PARTITION (Default is: null):
Enter path and name of KDC_PARTITIONFILE (Default is:
/opt/IBM/ITM/tables/REMOTE_EDINBURG/partition.txt):
```

Note: When IBM Tivoli Monitoring components need to communicate across a firewall that performs NAT, those components must be able to retrieve an IP address of the other component that is valid on its side of the firewall. To support this capability, the location broker namespace is logically divided into partitions with unique partition IDs. Partition IDs are specified using the KDC_PARTITION environment variable. The partition file is the means to insert appropriate IP addresses into the location broker namespaces.

7. If you want to use Configuration Auditing, type y and press Enter. Otherwise, press Enter. (During the installation, we chose to audit the configuration.)
8. Press Enter to accept the default setting for Hot Standby (NO).

   Tip: It is a good idea to wait until after you have fully deployed your environment to configure Hot Standby for your monitoring server. See 3.2.15, “Configuring the Hot Standby” on page 162.

9. Press Enter to accept the default for the Optional Primary Network Name (none).
10. .Press Enter for the default security: Validate User setting (no). If you need to use security validation in your environment, you can enable it after initial configuration is complete. See “Configuring user security” on page 69 for more information.

    Note: If you enable the security validation, you must create on the server the users who need to access the IBM Tivoli Monitoring 6.1 environment.

11. If you will be using event synchronization to view situation events, type y and press Enter to enable TEC Event Integration. Complete the following additional steps:
    a. Type the name of the IBM Tivoli Enterprise Console event server and press Enter.
    b. Type the port number for the event server and press Enter (if your event server is using port mapper; for example on a UNIX server, the port is 0. or if your event server is on a Windows machine the default port is 5529).

Note: If your Tivoli Enterprise Console is not yet set up, you can skip the previous section by typing NO and perform the configuration later using the same procedure.

c. Press Enter to not disable the Workflow Policy/Tivoli Emitter Agent.

12. Type `s` to save the SOAP configuration and exit the configuration. We will perform the SOAP configuration later, when configuring the event synchronization.

Installing agents support on (seeding) the hub monitoring server

After you have configured the TEMS, you must install the application support. Use the following steps to seed the hub monitoring server. Remember that seeding adds product-specific data to the monitoring server.

1. Start the monitoring server by running the following command from the $CANDLEHOME\bin directory:

   ```
   ./itmcmd server start <tems_name>
   ```

2. Run the following command to start the seeding process:

   ```
   ./itmcmd -t <tems_name> <pc pc pc>
   ```

   In this command, *tems_name* is the Hub TEMS name and *pc* is the product code for each agent whose data you want to send to the monitoring server. (You can retrieve the information by executing the command **`cinfo -i`**).

   Note: Seed the server with all available pc codes in advance.

3. Stop the monitoring server by running the following command:

   ```
   ./itmcmd server stop <tems_name>
   ```

4. Restart the monitoring server by running the following command:

   ```
   ./itmcmd server start <tems_name>
   ```

When you are done with the Hub TEMS, refer to Table 3-12 for installing the rest of the components.

Table 3-12 How to install IBM Tivoli Monitoring 6.1 components in scenario 2

Components	References
Install and configure the second UNIX Hub TEMS.	“Installing and configuring the scenario 2 environment” on page 169
Install and configure the remote TEMS.	“Installing a Remote TEMS on a Windows and UNIX server” on page 82
TEPS	“Tivoli Enterprise Portal Server - TEPS” on page 87
TEMAs	“Tivoli Enterprise Monitoring Agent” on page 95 and “Deploying TEMA from command line interface” on page 124
TEP	“Tivoli Enterprise Portal (TEP)” on page 126
Warehouse Proxy agent	“Warehouse Proxy installation and configuration” on page 130
Summarization and Pruning agent	“Summarization and Pruning agent installation and configuration” on page 139
Event synchronization	“Event synchronization installation” on page 147
Hot standby	“Configuring the Hot Standby” on page 162

3.2.17 Replacing a Hub TEMS server with a new one

For any reason (hardware upgrade, for example) if you want to replace your Hub TEMS server with a new one from the same platform or a different platform, IBM Tivoli Monitoring 6.1 allows you to do so. This section describes the procedure we followed to replace our scenario 1 implementation (Windows TEMS servers) with scenario 2 (UNIX TEMS servers).

1. Install, configure, and seed the Hub TEMS on the madrid and milan servers. Depending on the platform you are using, refer to “Installing and configuring a Hub TEMS on a Windows server” on page 69 or “Installing a Hub TEMS on a UNIX server” on page 169.
2. Stop all Remote TEMS (copenhagen and edinburg) using either the Manage Tivoli Enterprise Monitoring Services console or, for UNIX servers, use the following command:

   ```
   itmcmd server stop tems_name
   ```

   tems_name is the of the TEMS host by the server your are stopping its TEMS service.

3. Stop the TEPS, Warehouse Proxy agent, and Summarization and Pruning agent from the Manage Tivoli Enterprise Monitoring Services console.
4. Stop the Hub TEMS (cairo and helsinki) using Manage Tivoli Enterprise Monitoring Services console or **`itmcmd`** command for UNIX servers as describe previously.
5. Configure the Remote TEMS (copenhagen and edinburg) to point to your new Hub TEMS (madrid and milan primary and Hot Standby TEMS, respectively). Use the Reconfigure option on the Manage Tivoli Enterprise Monitoring Services console on UNIX servers or the following procedure on UNIX servers:

   ```
   itmcmd config -S -t tems_name
   ```

6. Configure the Warehouse Proxy agent and Summarization and Pruning agent to point to the primary and Hot Standby TEMS (madrid and milan) respectively. This operation is performed using the Manage Tivoli Enterprise Monitoring Services console.
7. Using the Manage Tivoli Enterprise Monitoring Services console, configure the TEPS to point to madrid.

Important: Before pointing TEPS to the new TEMS, you must save TEPS data using the migrate-export.bat facility, which detects the TEPS database and creates the SQL file script that will be used to restore your TEPS services. Use the following procedure to back up and import the data:

- Saving TEPS data

  Use the following procedure to back up your TEPS database:

  - **`cd`** to %CANDLEHOME%/CNPS
  - Run **`migrate-export.bat`**

  The migrate-export.bat facility stops the TEPS, creates a saveexport.sql, and restarts the TEPS.

- Importing backup data

  Use the following procedure to restore your TEPS initial state:

  - Import the data back into TEPS by copying the presentation files back to the CNP directory.
  - Copy the saveexport.sql file to the %CANDLEHOME%\cnps\sqllib directory and run migrate_import.bat.

  If you are using Linux as your TEPS, read $CANDLEHOME instead of %CANDLEHOME%.

8. Configure cairo to become a Remote TEMS. This is done using the Manage Tivoli Enterprise Monitoring Services console or itmcmd on a UNIX server. This step can be skipped if you do not want to reconfigure any of your old Hub TEMS to a Remote TEMS.
9. Configure agents on oslo and dakar to point to cairo and copenhagen, the primary and secondary Remote TEMS respectively. You do not have to reconfigure your agents to point to the new Hub; this operation was done to check whether the remote TEMS cairo is working as expected.
10. Start the Remote TEMS (edinburg, copenhagen, and cairo). Use the Manage Tivoli Enterprise Monitoring Services console or the `itmcmd` command for UNIX servers.
11. From the Manage Tivoli Enterprise Monitoring Services console, start the Warehouse Proxy agent, Summarization and Pruning agent, and TEPS.

Notes:

- This procedure is recommended only if you want to change your platform's Hub TEMS. If you want to build a two-Hub TEMS environment, you can install it from scratch.
- We were able to get all of the agents (from scenario 1) back up and running; we had to restart a few agents to have them reconnected. Others reconnected automatically.
- We lost our situations, workspace, and collections configuration because we did not save it to transfer it later on the new Hub TEMS.

Tips:

- If you want to change your Hub TEMS environment, back up the entire environment and plan the transition very carefully.
- If you are replacing one Hub TEMS with another one with the same platform and configuration (host name, TCP/IP configuration), you can back up the CANDLEHOME directory from the current Hub TEMS and restore it later onto the new Hub TEMS.
- If the new and old TEMS are installed on the same platform (such as both UNIX or Windows). you can back up your candle database from your current Hub TEMS and restore it later onto the new Hub TEMS. You must copy the following files from your old HUB TEMS.

  Windows:

  %CANDLEHOME%/CMS/QA1*.db and QA1*.idx

  UNIX:

  $CANDLEHOME/tables/TEMS_NAME/QA1*.db and QA1*.idx

3.3 Uninstalling IBM Tivoli Monitoring 6.1

In this section we uninstall IBM Tivoli Monitoring 6.1, both the whole environment and individual components.

3.3.1 Uninstalling the entire IBM Tivoli Monitoring environment

Use these procedures to remove the entire IBM Tivoli Monitoring environment.

Uninstalling the environment on Windows

Use the following steps to uninstall IBM Tivoli Monitoring from a Windows computer:

1. From the desktop, click **Start** → **Settings** → **Control Panel** (for Windows 2000) or **Start** → **Control Panel** (for Windows 2003).
2. Click **Add/Remove Programs.**
3. Select **IBM Tivoli Monitoring** and click **Change/Remove**.
4. Select **Remove** and click **Next**.
5. Click **OK**.
6. After Tivoli Enterprise services have stopped, you are asked if you want to remove the Tivoli Enterprise Portal database. Click **Yes**.

7. Type the password for the DB2 administrator in the Admin Password field and click **OK**.

   A pop-up window, indicating that GSKit is being uninstalled, is displayed.

8. Select **Yes** to restart your computer and click **Finish**.

Uninstalling the environment on UNIX

Before executing the uninstallation procedure, make sure that all IBM Tivoli Monitoring 6.1 components are shut down.

1. Stop the agent by executing this command from $CANDLEHOME/bin:

   ```
   ./itmcmd agent stop pc
   ```

   pc is the product code (lz, ux, ul, um, ui, cj, and so on).

2. Stop the TEMS by executing this command from $CANDLEHOME/bin:

   ```
   ./itmcmd server stop TEMS_NAME
   ```

3. Run the following command:

   ```
   ./uninstall.sh
   ```

   A numbered list of product codes, architecture codes, version and release numbers, and product titles is displayed for all installed products.

4. Type the number for the installed product that you want to uninstall. Repeat this step for each additional installed product you want to uninstall (Example 3-24).

Example 3-24 Uninstalling the environment on UNIX

```
*********** Mon Oct 10 15:08:29 EDT 2005 ******************
User      : itmuser      Group: itmuser
Host name : edinburg.itsc.austin.ibm.com        Installer Lvl: 400 / 100
CandleHome: /opt/IBM/ITM
***********************************************************

...Products available to uninstall

Num     Product [ Code  Platform Version:Release Description ]
1       cj  li6243      v610:r172  Tivoli Enterprise Portal Desktop Client
2       lz  li6263      v610:r115  Monitoring Agent for Linux OS
3       ms  li6243      v610:r215  Tivoli Enterprise Monitoring Server
4       sh  li6243      v610:r215  Tivoli Enterprise Monitoring SOAP Server
5       uf  li6243      v610:r100  Universal Agent Framework
6       ui  li6243      v610:r194  Tivoli Enterprise Services User Interface
7       um  li6243      v610:r229  Universal Agent

Enter number for a product to uninstall or "EXIT" to exit: 1
```

```
Confirm: cj  li6243  v610:r172 Tivoli Enterprise Portal Desktop Client ... OK
to delete? [y/n]: y
```

5. When finished, restart the computer to complete the uninstallation.

Notes:

- If for any reason the UNIX uninstallation is not successful, run the following command to remove all IBM Tivoli Monitoring directories:

  ```
  rm -r $CANDLEHOME
  ```

- If you are uninstalling the components using a user ID different from root user you might have errors like the following:

  ```
  rm: cannot remove `/etc/rc0.d/K10ITMAgents1': Permission denied
  rm: cannot remove `/etc/rc1.d/K10ITMAgents1': Permission denied
  rm: cannot remove `/etc/rc2.d/S99ITMAgents1': Permission denied
  rm: cannot remove `/etc/rc3.d/S99ITMAgents1': Permission denied
  rm: cannot remove `/etc/rc4.d/S99ITMAgents1': Permission denied
  rm: cannot remove `/etc/rc5.d/S99ITMAgents1': Permission denied
  rm: cannot remove `/etc/rc6.d/K10ITMAgents1': Permission denied
  rm: cannot remove `/etc/init.d/ITMAgents1': Permission denied
  ```

Contact your system administrator to remove those files.

3.3.2 Uninstalling an individual agent or component

Use the following procedures to remove an agent or other individual IBM Tivoli Monitoring component from your computer.

Uninstalling a component on Windows

Use the following steps to remove a component on a Windows computer. You can uninstall a single agent or the entire agent bundle (such as IBM Tivoli Monitoring for Databases).

1. From the desktop, click **Start** → **Settings** → **Control Panel** (for Windows 2000) or **Start** → **Control Panel** (for Windows 2003).
2. Click **Add/Remove Programs**.
3. Do one of the following:
   - To uninstall a single IBM Tivoli Monitoring component, such as the portal server or portal client (but not all components), select **IBM Tivoli Monitoring**.
   - To uninstall an agent bundle or a specific agent, select the agent bundle.
4. Click **Change/Remove**.

5. Do one of the following:
   - To uninstall a specific agent or component, select **Modify**.
   - To uninstall the entire agent bundle, select **Remove**.
6. Click **Next**.
7. Do one of the following:
   - If you are uninstalling an agent bundle, click **OK** to confirm the uninstallation.
   - If you are uninstalling an agent or component, do the following:
     i. For an agent, expand **Tivoli Enterprise Monitoring Agents** and select the agent you want to uninstall.
     ii. For a component, select the component (such as Tivoli Enterprise Portal Desktop Client).
     iii. Click **Next**.
     iv. Click **Next** on the confirmation screen.
     v. Depending on the remaining components on your computer, there might be a series of configuration panels. Click **Next** on each.
8. Click **Finish** to complete the uninstallation.
9. Restart the computer to complete the uninstallation.

Note: When removing a specific component (Modify/Remove), do not unselect any component other than the one you are removing. Deselecting any other component will uninstall it from the machine.

Uninstalling a component on UNIX

Use the following steps to remove a component from a UNIX computer. You can uninstall a single agent or the entire agent bundle (such as IBM Tivoli Monitoring for Databases).

1. From a command prompt, run the following command to change to the appropriate /bin directory:

   ```
   cd $CANDLEHOME/bin
   ```

   $CANDLEHOME is the path for the home directory for IBM Tivoli Monitoring.
2. Run the following command:

   ```
   ./uninstall.sh
   ```

   A numbered list of product codes, architecture codes, version and release numbers, and product titles is displayed for all installed products.

3. Type the number for the agent or component that you want to uninstall. Repeat this step for each additional installed product you want to uninstall.
4. Restart the computer to complete the uninstallation.

3.3.3 Uninstalling the TEC event synchronization

Use the following steps to uninstall the event synchronization from your event server:

1. Stop event synchronization on the event server by running the following command:

   On Windows:

   ```
   <tec_installdir>\TME\TEC\OM_TEC\bin\stop.cmd
   ```

   On UNIX:

   ```
   <tec_installdir>/TME/TEC/OM_TEC/bin/stop.sh
   ```

2. Run the following uninstallation program:

   On Windows:

   ```
   <tec_installdir>\TME\TEC\OM_TEC\_uninst\uninstaller.exe
   ```

   On UNIX:

   ```
   tec_installdir/TME/TEC/OM_TEC/_uninst/uninstaller.bin
   ```

   tec_installdir is the location of the IBM Tivoli Enterprise Console installation.

3. Follow the prompts in the uninstallation program.

Notes:

- You can also run this uninstallation program in silent mode (by running the program from the command line with the -silent parameter) or in console mode (by using the -console parameter).
- You must stop and restart the event server for these changes to take effect.
- If your event server is running on an HP-UX computer, ensure that the _uninst and _jvm directories are successfully removed by the uninstallation program. If they are not, manually delete these directories.

4

Historical summarized data

This chapter describes the architecture, planning, and implementation of IBM Tivoli Monitoring 6.1 Historical Data Collection. One of the primary features of the new IBM Tivoli Monitoring 6.1 product is the new Tivoli Data Warehouse (TDW) 2.1. In this chapter we discuss the overall architecture of how historical data is collected on IBM Tivoli Monitoring 6.1 agents, how historical data is collected by a new warehouse server in Tivoli Data Warehouse V2.1, and how historical summarization and pruning occurs within the new Tivoli Data Warehouse V2.1 database. We explain the details of the architecture of IBM Tivoli Monitoring 6.1 Historical Data Collection and Tivoli Data Warehouse V2.1, and show how Tivoli Data Warehouse V2.1 differs from the previous Tivoli Data Warehouse V1.x architecture.

Although the new Tivoli Data Warehouse V2.1 might be used in the future with other products, in this chapter we focus only on the IBM Tivoli Monitoring 6.1 historical data.

This chapter also uses real-world scenarios to plan, design, and configure IBM Tivoli Monitoring 6.1 and Tivoli Data Warehouse V2.1 historical collection. The scenarios include the configuration of the historical data for the following agents: Windows OS monitoring, Linux OS monitoring, AIX OS monitoring, and DB2 on UNIX monitoring. We also discuss some of the reporting tools that can be used with Tivoli Data Warehouse V2.1 database. Finally, we discuss how IBM Tivoli Monitoring 5.x data is integrated into the new Tivoli Data Warehouse V2.1.

We discuss the following topics in this chapter:

- Overview
- Architecture
- Planning
- Configuration
- Reporting

4.1 Overview

The new Tivoli Data Warehouse V2.1 has two new processes that collect, summarize, and prune data gathered from IBM Tivoli Monitoring 6.1 agents:

A new Warehouse Proxy agent is the new data warehouse server. The Warehouse Proxy agent collects data from the IBM Tivoli Monitoring 6.1 agents and stores the data in a relational database (DB2, Oracle, or MSSQL).

The Tivoli Data Warehouse V2.1 data can optionally be configured to summarize and prune the historical data with another new process called the Summarization and Pruning agent.

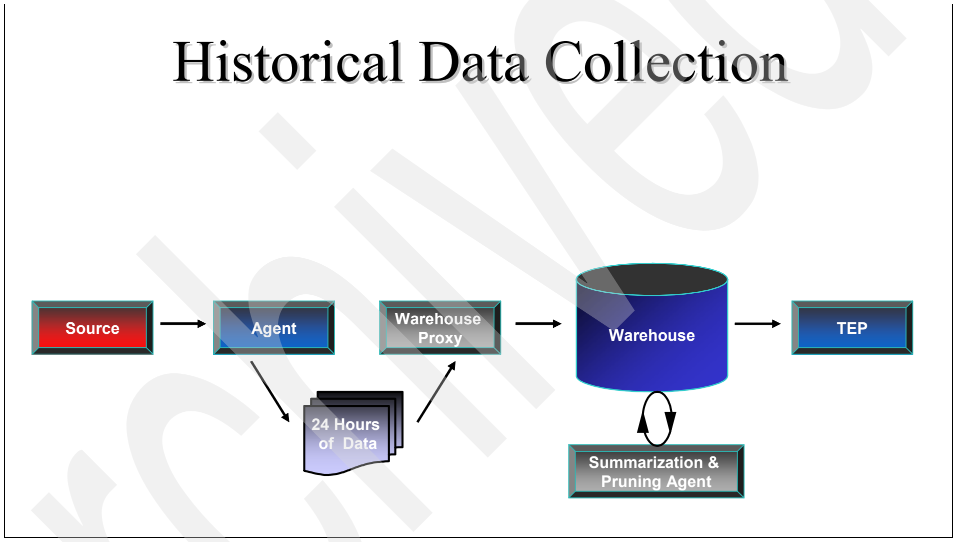


Figure 4-1 Historical data collection overview

These new processes are discussed in more detail in 4.2, “Architecture” on page 191. In this overview section we focus on the primary differences between Tivoli Data Warehouse V2.1 and Tivoli Data Warehouse V1.x. The new Tivoli Data Warehouse V2.1 architecture is extremely different from the older Tivoli Data Warehouse V1.x solution. The primary differences between the two versions are discussed in the following three topics:

- Implementation differences
- Usability differences
- Scalability differences

4.1.1 Implementation differences

Summary of the implementation differences:

- Uses row-based schema versus star-based schema.
- Detailed data is now stored in the Tivoli Data Warehouse V2.1.
- Tivoli Data Warehouse V2.1 supports Oracle, MSSQL, and DB2.

One of the primary architectural differences between Tivoli Data Warehouse V2.1 and Tivoli Data Warehouse V1.x is that Tivoli Data Warehouse V2.1 is based on a single database and a simple row-based schema. Version 1.x was based on a multi-tiered database structure with a central data warehouse database and a separate datamart database, and it required a more complex process to load the databases using an ETL1 and ETL2 process.

The Tivoli Data Warehouse V2.1 architecture has a much simpler data collection and summarization architecture. The datamart database in Tivoli Data Warehouse V1.x was based on a star schema. A star schema is a series of fact and dimension tables that are arranged as a conceptual star. At the core of a star schema is a technique called *database normalization*. For more information, see:

http://en.wikipedia.org/wiki/Database_normalization

Row-based schemas are much simpler and are basically stored as a "what you see is what you get" arrangement. Row-based schema are easier to manipulate than star schemas. However, star schemas are far more efficient for processing large amounts of data. We cover this in 4.1.3, "Scalability differences" on page 190 and in more detail in 4.3, "Planning" on page 209.

Another difference between Tivoli Data Warehouse V2.1 and Tivoli Data Warehouse V1.x is that Version 2.1 now stores detailed level monitoring data (RAW data) in the data warehouse. The V1.x data was always aggregated to the hourly level when it was stored in the data warehouse. Version 2.1 also improves the way data aggregation occurs: This version is more flexible in ways attributes can be aggregated, and it always aggregates from the raw data (the detailed tables). This topic is discussed in more detail in 4.2, "Architecture" on page 191 in this chapter.

Finally, the Tivoli Data Warehouse V2.1 database is not restricted to DB2 as the older version was. Version 2.1 now support DB2 8.2 or higher, Oracle 9 or higher, and MS SQL Server 2000.

4.1.2 Usability differences

Summary of the usability differences:

- Tivoli Data Warehouse V2.1 stores more than 24 hours of detailed data.
- Detailed and summarized data reside in the same database.
- New GUI (Tivoli Enterprise Portal) access to Tivoli Data Warehouse.
- An improved time-to-value.

The usability differences between Tivoli Data Warehouse V2.1 and Tivoli Data Warehouse V1.x are significant. One of the biggest new features with Version 2.1 is that customers now can access more than 24 hours of detailed data. Users can configure a maximum time to keep detailed data that is architecturally unlimited. (Planning considerations should apply.) For example, a customer can now keep 30 days of detailed data in Tivoli Data Warehouse V2.1 if they choose (see "Planning" on page 209). Also in Version 2.1, the detailed data is stored in the same database as the summarized data, so users no longer have to navigate to multiple interfaces to run reports, as all of the data is in the same database and can be accessed through a common portal. Access to all levels of data is vastly improved in Version 2.1. In Tivoli Data Warehouse V1.x a user had to use the Web Health Console to get to the detailed data and use a reporting tool such as Alphabox, Crystal Reports, Brio, or SAS to access the summarized data (the datamart). In Tivoli Data Warehouse V2.1 a customer can access all tables (detailed or summarized data) from the same repository using any RDBMS reporting tool or using the Tivoli Enterprise Portal.

The IBM Tivoli Monitoring 6.1 new portal provides a very robust GUI. The Tivoli Enterprise Portal (TEP) interface is used for querying and viewing all levels of the stored historical data. IBM Tivoli Monitoring 6.1 provides seven agents out of the box with thousands of out-of-the-box queries. The fact that the combination of IBM Tivoli Monitoring 6.1 and Tivoli Data Warehouse V2.1 gives the customer seamless access to all levels of the data is a major improvement over the Tivoli Data Warehouse V1.x architecture.

Finally, IBM Tivoli Monitoring 6.1 and Tivoli Data Warehouse V2.1 provide an improved time-to-value proposition. IBM Tivoli Monitoring 6.1 and Tivoli Data Warehouse V2.1 can be implemented in a matter of hours compared to a few days and more likely weeks with the old Tivoli Data Warehouse V1.x. Obviously more than a few hours should be dedicated to the planning of a Tivoli Data Warehouse V2.1 implementation and we discuss this in more detail in "Planning" on page 209.

4.1.3 Scalability differences

Summary of the scalability differences:

- Data collection
- Planning
- Performance

The new Tivoli Data Warehouse V2.1 features of storing detailed data in the database and allowing more than 24 hours of data can provide a tremendous opportunity. However, these features can also create some new pitfalls if proper planning is not applied. In this chapter we discuss some of the implementation and usability weaknesses that were encountered using IBM Tivoli Monitoring 5.x with Tivoli Data Warehouse V1.x.

However, the IBM Tivoli Monitoring 5.x - Tivoli Data Warehouse V1.x architecture handled data collection and retrieval very well. In the Tivoli Data Warehouse V1.x architecture, data was collected and retrieved by IBM Tivoli Monitoring 5.x from the endpoints using the Tivoli Framework MDist2 architecture. When a user requested a Web Health Console screen that required detailed information, a request was made to the gateway and then to the endpoint using MDist2. All data collections from the endpoints in IBM Tivoli Monitoring 5.x are initiated from the endpoint's gateways and the data is uploaded into an RDBMS from a gateway. This data collection process also makes use of the Framework's MDist2 architecture.

IBM Tivoli Monitoring 6.1 does not use the Tivoli Framework architecture and MDist2 to collect data. In Version 6.1, if a user does a query against detailed data for less than 24 hours for multiple servers, the overhead of the request could significantly affect the performance of the overall IBM Tivoli Monitoring 6.1 monitoring infrastructure. In contrast, IBM Tivoli Monitoring 5.x and the MDist2 architecture always retreived and collected data using a multi-tier structure (for example, endpoint- → gateway → RDBMS). MDist2 also has a very robust architecture for throttling and tuning data movement throughout an infrastructure.

Probably a more important scalability difference is that customers can now collect an enormous amount of data in Tivoli Data Warehouse V2.1. The collection and reporting against this data can have an impact on the IBM Tivoli Monitoring 6.1 monitoring infrastructure. In Tivoli Data Warehouse V1.x, all of the warehouse data was already aggregated to the hourly level on the endpoint and only the aggregated data was uploaded to the gateways. The new Tivoli Data Warehouse V2.1 architecture uses one server (called the Warehouse Proxy agent) that collects detailed data directly from all endpoints. This one server initiates requests from all of the agents to export data directly to the Warehouse Proxy agent using ODBC connections.

User reporting can also negatively affect monitoring performance, depending on the configuration settings. In the Tivoli Data Warehouse V1.x architecture, the datamart was in a separate database from the operational data (the IBM Tivoli Monitoring RIM database). Now in Tivoli Data Warehouse V2.1, a single database is both the operational database and the data warehouse database. In Version 2.1, multiple users can run the same report against hundreds of servers for millions of lines of data. In Tivoli Data Warehouse V1.x the data was summarized and also normalized using datamarts and star schemas. Users might want to consider using an OEM tool to create OLAP cubes or datamarts in Tivoli Data Warehouse V2.1 to offload some of the reporting requirements.

4.2 Architecture

In this section we discuss the architecture of Tivoli Data Warehouse V2.1 as it relates to IBM Tivoli Monitoring 6.1 monitored data. The main topics that are discussed in this section are:

- Historical data architecture overview
- Historical data types
- Component flows
- Data tables and attributes
- Object definitions

4.2.1 Historical data architecture overview

The IBM Tivoli Monitoring 6.1 Historical Data Collection architecture is comprised of three primary components. The following components are used to collect data in the IBM Tivoli Monitoring 6.1 architecture:

- Warehouse Proxy agent
- Summarization and Pruning agent
- Tivoli Data Warehouse V2.1 (Tivoli Data Warehouse V2.1)

The Warehouse Proxy agent

The Warehouse Proxy agent is the bridge between the active monitoring system and the historical data repository. It handles warehousing requests from all managed systems in the enterprise and uses ODBC to write the historical data to a supported relational database. Only one Warehouse Proxy agent at a time can be configured and running in an IBM Tivoli Monitoring instance (Hub TEMS). The Warehouse Proxy can only connect successfully to a hub monitoring server.

Summarization and Pruning agent

The Summarization and Pruning agent maintains the data within the data warehouse by aggregation and pruning data based on customer specifications.

The IBM Tivoli Monitoring administrator sets up how often to collect the detailed data, what intervals to aggregate and prune on, and how often to run the aggregation and pruning engine. Typically the summarization and running process is scheduled to run once a day.

Tivoli Data Warehouse V2.1

The Tivoli Data Warehouse database is an integral part of the solution. It stores a large amount of attribute data and customers will want to host this data on existing database farms. The database will be used by the TEP if historical data is to represented. External reporting tools and other applications can access the data and operate off of this database.

4.2.2 Historical data types

There are two types of data stores for the IBM Tivoli Monitoring 6.1 Historical Data Component.

- Short-term data
- Long-term data

Short-term data

Short-term data is typically referred to in IBM Tivoli Monitoring 6.1 as data that is stored in binary files and is less than 24 hours old. In the IBM Tivoli Monitoring 6.1 architecture, short-term data can be configured to store the binary files locally on the Tivoli Enterprise Management Agent (TEMA) or it can be configured to store the binary files on the Tivoli Enterprise Monitoring Server (TEMS). This can be configured by a user by agent type. In both cases (TEMA or TEMS) the binary data is considered short-term because it is only designed for 24-hour access. When the Summarization and Pruning agent is configured it can be set up to prune this short-term data. When the short-term data is successfully loaded into the Tivoli Data Warehouse V2.1 to the Warehouse Proxy agent, it is pruned on TEMA or TEMS if it is older than 24 hours. If the Warehouse Proxy agent is not configured to collect the short-term data, then a user-defined pruning job must be implemented. When we wrote this book, and unless otherwise specified by the specific agent, it was recommended that the location for the binary short-term data be located the TEMA. This configuration option is discussed in 4.4, “Configuration” on page 224. The binary short-term data will never be in aggregate or summarized format regardless of whether it is stored on the TEMA or the TEMS.

Long-term data

Long-term data in IBM Tivoli Monitoring 6.1 is typically referred to as data that is older than 24 hours and has been collected up to the Tivoli Data Warehouse V2.1 RDBMS to the Warehouse Proxy agent. The long-term data resides in tables in the Tivoli Data Warehouse V2.1 database. The long-term RDBMS tables contain detailed data and summarized data in the Tivoli Data Warehouse V2.1. The Summarization and Pruning agent can be configured to run every day to roll up data from the detailed level to hourly, to weekly, to monthly, to quarterly, to yearly. The Summarization and Pruning agent also prunes the summarized tables (Figure 4-2).

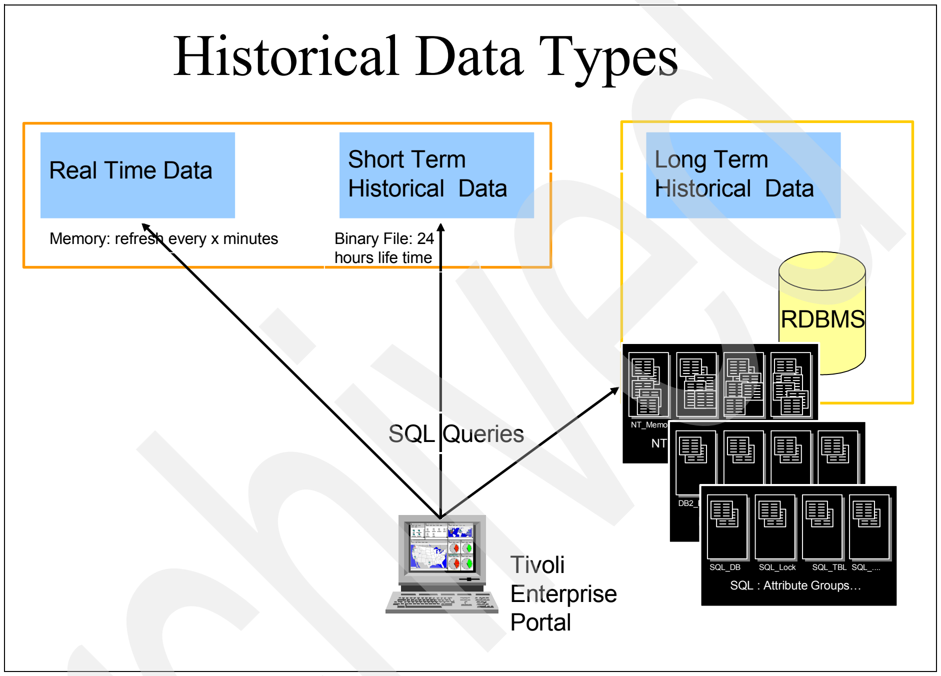


Figure 4-2 Historical data types

4.2.3 Component flows

When Historical Data Collection is configured in IBM Tivoli Monitoring 6.1, a user can determine whether the short-term data (the binary 24-hour data) should be stored on the Tivoli Enterprise Monitoring Server (TEMS) or on the Tivoli Enterprise Monitoring Agent (TEMA). If the data is stored on the TEMA then each monitored machine will store binary files for all of the monitoring agents running on that system. As we wrote this book, collecting historical data on the TEMS for large-scale customers was not recommended. “Planning” on page 209 discusses this in more detail.

In some cases it might be necessary to collect short-term historical data on the TEMS. Some agents require this configuration; it also might be necessary if there are firewall considerations. If the short-term Historical Data Collection is configured to collect on the TEMS, the binary files for all monitored machines and their agents will be collected up to the TEMS. This will create a single binary file for each type of monitoring attribute group for all machines and can become a single point of failure and cause reporting queries to run for a long time.

When the Warehouse Proxy agent and Summarization and Pruning agent are installed and configured, data is loaded from the TEMA or TEMS (depending on the location setting) to the Tivoli Data Warehouse V2.1 RDBMS. When data is collected to the Warehouse Proxy agent, tables are created in the Tivoli Data Warehouse V2.1 database. When the historical collection is configured a user can specify how often to prune the raw data. The default is seven days. After the raw data has been loaded into the Tivoli Data Warehouse V2.1 tables, data older than 24 hours will be pruned from the short-term binary files located on the TEMA or TEMS. At any given time you could have 24 hours of short-term raw data on the TEMA or TEMS and detailed tables in the Tivoli Data Warehouse RDBMS that contains the same data. When a request is made from the Tivoli Enterprise Portal (TEP) to perform a query that uses the timespan function, data is retrieved from the binary file if the timespan is less than or equal to 24 hours. A query performed from the TEP that uses a timespan greater than 24 hours will retrieve data from the Tivoli Data Warehouse V2.1 tables.

Important: The most recent 24 hours' worth of data comes from a binary file stored at the agent or at the Tivoli Enterprise Monitoring Server. Beyond 24 hours, the data is retrieved from the Tivoli Data Warehouse. The TEMS determines where to get the data: either from the agent, if the data is less than 24 hours old, or from the Tivoli Data Warehouse, if the data is older than 24 hours. If the query goes to an agent and retrieves a large amount of data, it can consume a large amount of CPU and memory. You can experience low system performance while a large amount of data is retrieved from the agents.

Figure 4-3 shows the flow of historical data collected when the location is stored on the TEMA.

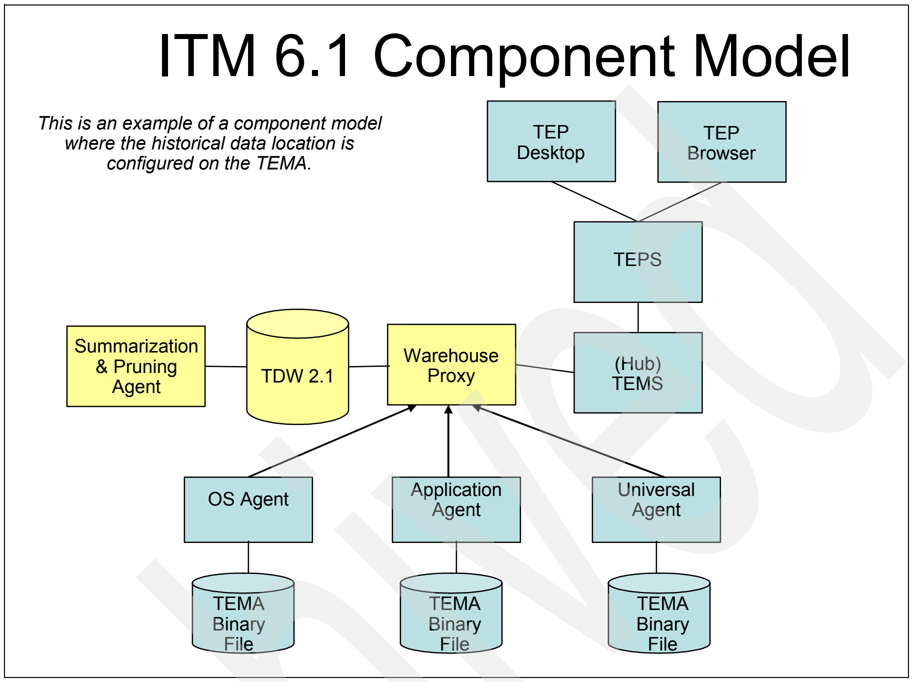


Figure 4-3 Historical collection location TEMA

Figure 4-4 shows the flow of historical data collected when the location is stored on the TEMS.

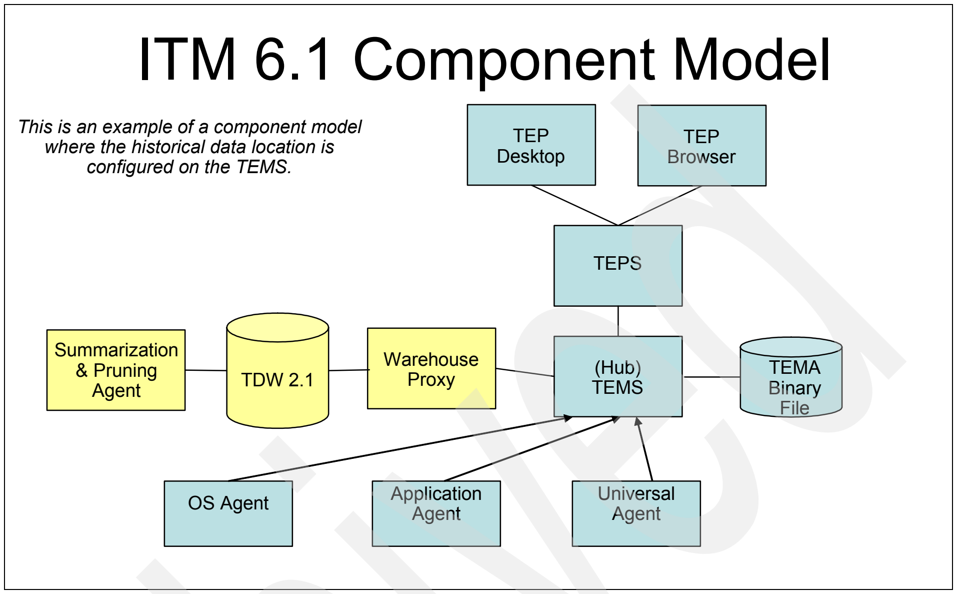


Figure 4-4 TM 6.1 Component model (historical collection location TEMS)

4.2.4 Data tables and attributes

Historical collection of data is based on *attribute groups*, which are defined as groupings of attributes within a specific IBM Tivoli Monitoring 6.1 agent. For example, the IBM Tivoli Monitoring 6.1 Monitoring Agent for Windows OS has 42 attribute groups with more than 1000 attributes. Each agent has a set of default attribute groups defined that can be configured easily for historical monitoring, Additional attribute groups can be configured if needed. There is a separate user guide for each supported IBM Tivoli Monitoring 6.1 agent that describes the agent's attribute groups and attributes.

Note: As we wrote this book, the shipped agent user guides were inconsistent regarding the documentation for describing the Tivoli Data Warehouse V2.1 schemas. Currently the only accurate way to get the schema table names is by querying the databases or looking at the Object Definition Interchange (ODI) files. ODI is described in 4.2.5, "Object definitions" on page 207.

Table 4-1 is an example of three IBM Tivoli Monitoring 6.1 Agents and their default attribute groups.

Table 4-1 Default attribute group examples

Agent	Default attribute group
Monitoring Agent for Windows OS	Network_Interface NT_Processor NT_Logical_Disk NT_Memory NT_Physical_Disk NT_Server NT_System
Monitoring Agent for UNIX	Disk System
Monitoring Agent for Linux	Linux_CPU Linux_CPU_Averages Linux_CPU_Config Linux_Disk Linux_Disk_IO Linux_Disk_Usage_Trends Linux_IO_Ext Linux_Network Linux_NFS_Statistics Linux_OS_Config Linux_Process Linux_RPC_Statistics Linux_Sockets_Status Linux_Swap_Rate Linux_System_Statistics Linux_User_Login Linux_VM_Stats
Monitoring Agent for DB2	KUDDBASEGROUP00 KUDDBASEGROUP01 KUDBUFFERPOOL00 KUDINFO00 KUDTABSPACE

Short-term binary tables

When Historical Data Collection is turned on in IBM Tivoli Monitoring 6.1, the default attribute groups can be configured to collect historical data (see 4.4, “Configuration” on page 224). After the data collection starts, the agent starts storing short-term binary tables on the TEMA or TEMS depending on the collection location that has been configured. For example, Table 4-2 on page 198 lists four agents' default binary file table names. These are the names of the binary file tables as they appear on the TEMA or TEMS.

Table 4-2 Short-term binary table names

Agent	Binary table name
Monitoring Agent for Windows OS	NETWRKIN NTPROCSSR WTLOGCLDSK WTMEMORY WTPHYSDSK WTSYSTEM
Monitoring Agent for UNIX	UNIXDISK UNIXOS
Monitoring Agent for Linux	LNXCPU LNXCPUAVG LNXCPUCON LNXDISK LNXDSKIO LNXDU LNXIOEXT LNXLOGIN LNXNET LNXNFS LNXOSCON LNXPROC LNXRPC LNXSOCKS LNXSWPRT LNXSYS LNXVM
Monitoring Agent for DB2	KUD3437500 KUD3437600 KUD4177600 KUD4238000 KUDTABSPC

Note: As we wrote this book, all of the tables listed in Table 4-1 on page 197 and Table 4-2 on page 198 were the shipped defaults. Some of the Linux default tables may change in the GA and post-GA releases of IBM Tivoli Monitoring 6.1. For example, Linux_Process should be removed from the default list because it can collect an enormous amount of data.

Each short-term binary file table also has an HDR file. Every binary file table has an associated HDR file (for example, NTPROCSSR.hdr). The timestamp of the HDR file can be useful to determine the first time that data collection took place for that attribute group. The timestamp on the tablename (that is, the file without the *.hdr) indicates the last time data collection occurred for that attribute group. Using the timestamps of these files can be helpful for troubleshooting purposes.

The short-term binary tables are not accessed directly by a user. The binary tables are only accessed from the TEP for queries of data less than 24 hours. The binary tables are also in a proprietary format. Although the tables cannot be acessed directly, it may be helpful to know the names of the tables to determine whether short-term historical data is being collected and for troubleshooting. The short-term tables can be found in the default IBM Tivoli Monitoring 6.1 install directory. For example:

For Windows:

```
C:\IBM\IBM Tivoli Monitoring\tmaIBM Tivoli Monitoring6\logs
```

For UNIX:

```
/opt/IBM/IBM Tivoli Monitoring/<platform abbreviation>/lz/hist
```

Examples of <platform abbreviation> are li6263 for Linux and aix513 for UNIX.

Note: The platform abbreviation varies based on product and platform support (such as between 32-bit and 64-bit).

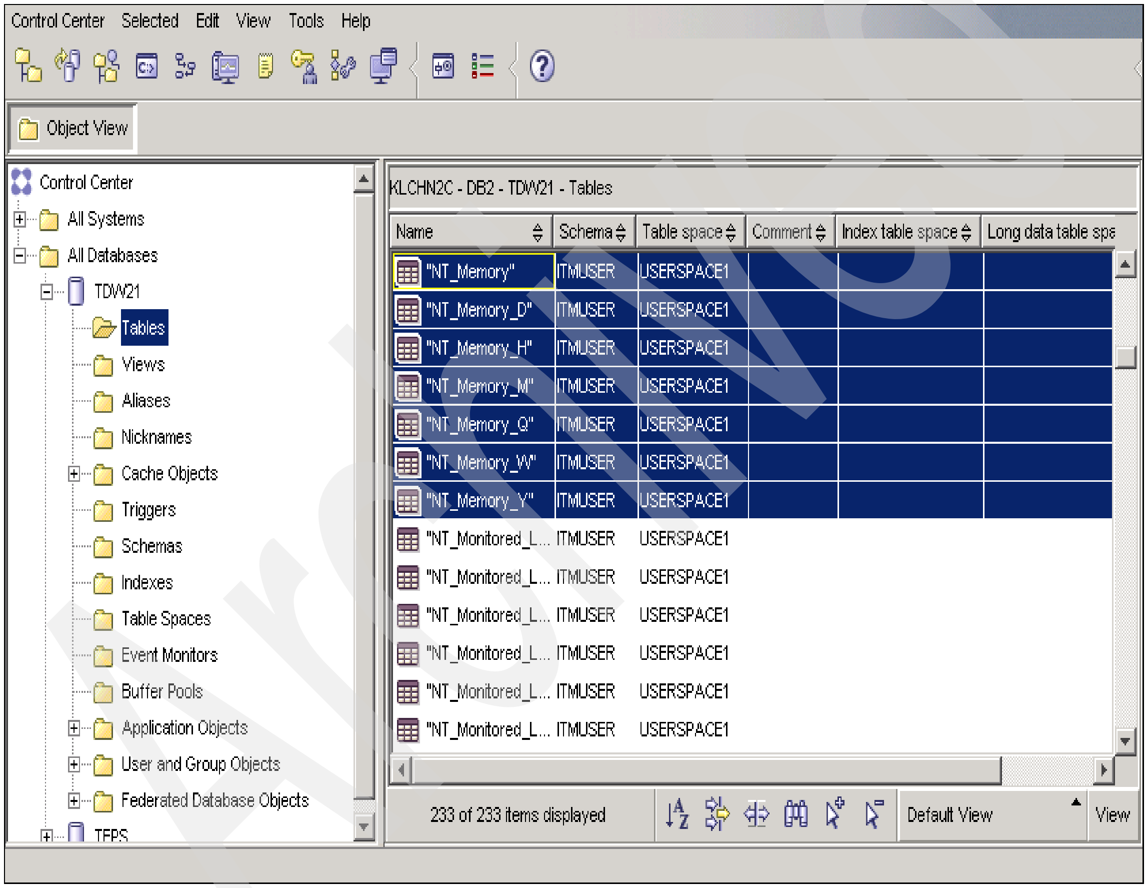
Long-term RDBMS tables

At the core of the Tivoli Data Warehouse V2.1 is a single RDBMS database. The Version 2.1–supported databases are DB2 8.2 or higher, Oracle 9.2 or higher, and MS SQL Server 2000. When an attribute group is configured and has started historical collection of data, a set of tables is created in the Tivoli Data Warehouse: one detailed table and multiple summarization tables for each attribute group. For example, if yearly, quarterly, monthly, weekly, daily, and hourly summarization are turned on for the NT_Memory attribute group, the following tables will be created in the Tivoli Data Warehouse:

“NT_Memory”	The detailed historical table for NT_Memory.
“NT_Memory_H”	The summarized hourly historical table for NT_Memory.
“NT_Memory_D”	The summarized daily historical table for NT_Memory.
“NT_Memory_W”	The summarized weekly historical table for NT_Memory.
“NT_Memory_M”	The summarized monthly historical table for NT_Memory.
“NT_Memory_Q”	The summarized quarterly historical table for NT_Memory.
“NT_Memory_Y”	The summarized yearly historical table for NT_Memory.

Note: All Tivoli Data Warehouse V2.1 table names are created with quotation marks surrounding the table name. When referencing historical data in the Tivoli Data Warehouse database, you must use the double-quote character to ensure correct access to that data.

Example 4-1 on page 208 shows how to use an SQL query to get a list of all table names. Figure 4-5 is a list of the NT_Memory tables in the Tivoli Data Warehouse from the DB2 8.2 Control Center.



KLCHN2C - DB2 - TDW21 - Tables

Name	Schema	Table space	Comment	Index table space	Long data table spa
"NT_Memory"	ITMUSER	USERSPACE1			
"NT_Memory_D"	ITMUSER	USERSPACE1			
"NT_Memory_H"	ITMUSER	USERSPACE1			
"NT_Memory_M"	ITMUSER	USERSPACE1			
"NT_Memory_Q"	ITMUSER	USERSPACE1			
"NT_Memory_W"	ITMUSER	USERSPACE1			
"NT_Memory_Y"	ITMUSER	USERSPACE1			
"NT_Monitored_L...	ITMUSER	USERSPACE1			
"NT_Monitored_L...	ITMUSER	USERSPACE1			
"NT_Monitored_L...	ITMUSER	USERSPACE1			
"NT_Monitored_L...	ITMUSER	USERSPACE1			
"NT_Monitored_L...	ITMUSER	USERSPACE1			
"NT_Monitored_L...	ITMUSER	USERSPACE1			

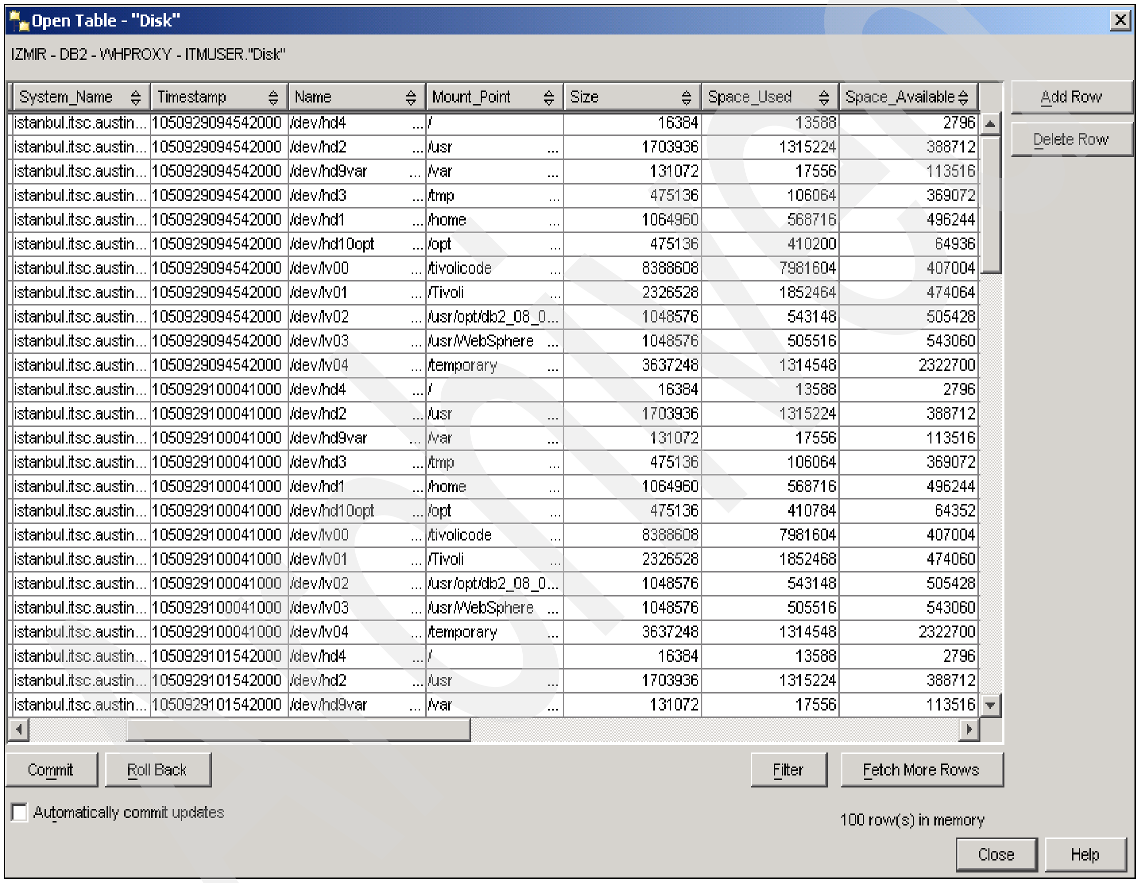
233 of 233 items displayed

Figure 4-5 Example of NT_Memory Detail and Summarization tables

Some attribute groups collect data for single-instance attributes and some attribute groups collect attributes for multiple-instance attributes. The NT_Memory attribute group is an example of a single-instance attribute group. The “NT_Memory” detailed table has only one row per collection interval. The

Monitoring Agent for UNIX attribute group for disk monitoring creates a table called “Disk”. The “Disk” attribute group collects data for UNIX file systems and is a good example of a multiple-instance attribute group. The “Disk” detailed table will have multiple rows per collection interval, with a row for each file system found on the specific agent.

Figure 4-6 shows an example of the UNIX Disk attribute group with collected data in the Tivoli Data Warehouse. Notice that the 1050929094542000 timestamp has eleven filesystems for that one collection (cycle).


Open Table - "Disk"

IZMIR - DB2 - WHPROXY - ITMUSER."Disk"

System_Name	Timestamp	Name	Mount_Point	Size	Space_Used	Space_Available
istanbul.itsc.austin...	1050929094542000	/dev/hd4 ...	/	16384	13588	2796
istanbul.itsc.austin...	1050929094542000	/dev/hd2 ...	/usr ...	1703936	1315224	388712
istanbul.itsc.austin...	1050929094542000	/dev/hd9var ...	/var ...	131072	17556	113516
istanbul.itsc.austin...	1050929094542000	/dev/hd3 ...	/tmp ...	475136	106064	369072
istanbul.itsc.austin...	1050929094542000	/dev/hd1 ...	/home ...	1064960	568716	496244
istanbul.itsc.austin...	1050929094542000	/dev/hd10opt ...	/opt ...	475136	410200	64936
istanbul.itsc.austin...	1050929094542000	/dev/lv00 ...	/tivolicode ...	8388608	7981604	407004
istanbul.itsc.austin...	1050929094542000	/dev/lv01 ...	/Tivoli ...	2326528	1852464	474064
istanbul.itsc.austin...	1050929094542000	/dev/lv02 ...	/usr/opt/db2_08_0...	1048576	543148	505428
istanbul.itsc.austin...	1050929094542000	/dev/lv03 ...	/usr/WebSphere ...	1048576	505516	543060
istanbul.itsc.austin...	1050929094542000	/dev/lv04 ...	/temporary ...	3637248	1314548	2322700
istanbul.itsc.austin...	1050929100041000	/dev/hd4 ...	/	16384	13588	2796
istanbul.itsc.austin...	1050929100041000	/dev/hd2 ...	/usr ...	1703936	1315224	388712
istanbul.itsc.austin...	1050929100041000	/dev/hd9var ...	/var ...	131072	17556	113516
istanbul.itsc.austin...	1050929100041000	/dev/hd3 ...	/tmp ...	475136	106064	369072
istanbul.itsc.austin...	1050929100041000	/dev/hd1 ...	/home ...	1064960	568716	496244
istanbul.itsc.austin...	1050929100041000	/dev/hd10opt ...	/opt ...	475136	410784	64352
istanbul.itsc.austin...	1050929100041000	/dev/lv00 ...	/tivolicode ...	8388608	7981604	407004
istanbul.itsc.austin...	1050929100041000	/dev/lv01 ...	/Tivoli ...	2326528	1852468	474060
istanbul.itsc.austin...	1050929100041000	/dev/lv02 ...	/usr/opt/db2_08_0...	1048576	543148	505428
istanbul.itsc.austin...	1050929100041000	/dev/lv03 ...	/usr/WebSphere ...	1048576	505516	543060
istanbul.itsc.austin...	1050929100041000	/dev/lv04 ...	/temporary ...	3637248	1314548	2322700
istanbul.itsc.austin...	1050929101542000	/dev/hd4 ...	/	16384	13588	2796
istanbul.itsc.austin...	1050929101542000	/dev/hd2 ...	/usr ...	1703936	1315224	388712
istanbul.itsc.austin...	1050929101542000	/dev/hd9var ...	/var ...	131072	17556	113516

Figure 4-6 UNIX “Disk” Table (multiple-instance) example

Example 4-1 on page 208 shows a detailed list of the Tivoli Data Warehouse V2.1 table names and instance types.

Note: When an attribute group is configured and collection is started, all of the definitions for that attribute group are common for all agents. In IBM Tivoli Monitoring 6.1 you cannot filter historical collection by agents or groups of agents. For example, if the NT_Memory attribute group is configured to collect historical data, then all Windows OS Agents will collect this attribute group. You cannot exclude certain machines or groups of machines for historical collection. Furthermore, all summarization and pruning definitions will be in effect for all agents that the attribute group apply to. In other words, if "NT_Memory" is configured to keep seven days of detailed data, there will be seven days of detailed data for all Windows machines that have the Windows OS Agent deployed.

Detailed tables

All of the detailed tables are based on a row-based schema. Each attribute group that has Historical Data Collection turned on creates its own unique table and unique columns. Attribute values in the detailed tables will store the actual raw values. Figure 4-6 on page 201 shows a display of the UNIX Disk table and some of the detailed values that are stored. All of the attribute groups and attributes are discussed in the IBM Tivoli Monitoring 6.1 specific agent monitoring guides. However, as we mentioned previously, there is no consistent documentation describing the table schemas that are available with IBM Tivoli Monitoring 6.1. One of the ways to get more information about tables and columns is discussed in 4.2.5, "Object definitions" on page 207.

Most of the columns in the detailed tables are unique according to their specific attribute group. However, three common columns are important to know in order to understand the Tivoli Data Warehouse V2.1 architecture and are useful for generating reports. These columns are:

- TMZDIFF

  The time zone difference from Universal Time (GMT). This value is shown in seconds.

- WRITETIME

  The time the record was written in the database. The format of this timestamp is a 16-character value in the format *cyymmddhhmmssttt*, where:

  – c = century

  – yymmdd = year, month, day

  – hhmmssttt = hours, minutes, seconds, milliseconds

- Timestamp

  This the date and time the agent collects information as set on the monitored system. The format of this timestamp is the same 16-character value (*cyymmddhhmmssttt)* used for WRITETIME.

The origin node field is another field that should be considered when working with the Tivoli Data Warehouse V2.1 architecture. The origin node is typically the host name of the resource and is different depending on the agent type.

Developers of agents should use certain general guidelines for the origin node field, but some agents do not follow those guidelines. In general the origin node is constructed as follows:

The originnode may be of this form: *instance*:*hostname*:*type*

- *instance* is optional.
- The delimiter usually is a colon.
- *hostname* is the machine name but it can also be a broker name (in case of MQ Series, for instance).
- *type* is the node type or product such as KNT for the Windows agent, KUX for the UNIX agent, and so on.

Here are some examples:

- Monitoring Agent for Windows OS

  The attribute for the monitored server name is Server_Name. For example:

  ```
  Primary:CAIRO:NT
  ```

- Monitoring Agent for UNIX OS

  The attribute for the monitored server name is Server_Name. For example:

  ```
  istanbul.itsc.austin.ibm.com:KUX
  ```

- Monitoring Agent for Linux OS

  The attribute for the monitored server name is Server_Name. For example:

  ```
  istanbul.itsc.austin.ibm.com:LZ
  ```

- Monitoring Agent for DB2

  The attribute for the monitored server name is Server_Name. For example:

  ```
  DB2:KLLAA9B:UD
  ```

One way to get a list of all table names in the Tivoli Data warehouse is to query the TEPS database. On the TEPS database machine or from an ODBC connection, you can run the following SQL statements to get a list of all the installed detailed tables from DB2:

```
connect to teps use db2admin using password
select tabname,longtable,product from teps.kfwhistdata
```

In these statements:

- `tabname` is the short-term binary file table name
- `longtable` is the detailed table name in the RDBMS
- `product` is the name of the associated agent

Summarized tables

If summarization is configured for an attribute group, then additional tables that include summarized data will be created in the Tivoli Data Warehouse. Summarization is the process of aggregating the detailed data into time-based categories, for example, hourly, daily, weekly, quarterly, and yearly. Summarizing data enables you to perform historical analysis of the data over time. Along with summarization parameters, pruning definitions can also be defined. The Summarization and Pruning agent will create the summarized tables and perform the pruning process to remove old data. The Summarization and Pruning agent can be configured to run a summarization and running process once a day. When the Summarization and Pruning agent process (for example, ksy610.exe on Windows) is started it runs as a process on the system OS (Windows, UNIX, or Linux). This process will sleep and wakes up every 5 minutes to check whether the summarization and pruning run has been scheduled to kick off. (The default schedule is once per day at 02:00 a.m.) If the summarization and pruning is scheduled to run within this 5 minute interval it will then start the summarization and pruning agent scheduled run against the Tivoli Data Warehouse V2.1 database. The summarization portion of the run is a rollup process that aggregates data from the detailed tables to the specific summarization time-based tables (hourly, daily, weekly, quarterly, and yearly). The pruning portion of the run will remove data from the detailed and summary tables based on the configured pruning parameters. The default pruning parameters are as follows:

- 7 days of raw data
- 90 days of hourly data
- 12 months of daily data
- 2 years of weekly data
- 3 years of monthly data
- 3 years of yearly data

Important: You can run only one Summarization and Pruning agent even if you have multiple Tivoli Enterprise Monitoring Servers that are sharing a single Tivoli data warehouse database. Running multiple Summarization and Pruning agents causes conflicts because the multiple instances would attempt to prune the data in the tables simultaneously. The negative impact is that the configuration settings for the summarization and pruning periods has to be set only in one Tivoli Enterprise Monitoring Server; that monitoring server controls how the data is summarized and pruned for all monitoring servers.

The names of the summarization tables will be the same as the detailed table name with an additional one-character identifier. Depending on the summarization interval that is chosen for the particular attribute group, the additional tables are created in the Tivoli Data Warehouse.

Table 4-3 Linux CPU tables example

Timespan	Example
Detail	Linux_CPU
Hourly	Linux_CPU_H
Daily	Linux_CPU_D
Weekly	Linux_CPU_W
Monthly	Linux_CPU_M
Quaterly	Linux_CPU_Q
Yearly	Linux_CPU_Y

The attributes in the summarized tables are stored in a different format than the detailed table attributes. When the attributed are aggregated in the summarized tables, they are stored in different formats depending on the type of data they represent. Eight aggregation behavior characterization types are used for aggregation; the following five types are used most often.

Behavior characterization types

- **GAUGE**

  Attributes that are range-based numeric data. These attributes are aggregated with MIN, MAX, AVG, and SUM values from the detailed data to the appropriate summarization period. There will be four attributes in the summarized table for each detailed attribute definition in the detailed table. The original attribute name is prefixed with MIN_, MAX_, AVG_, and SUM_.

For example the "Linux_CPU_D" table would have the following attributes for the System_CPU attribute:

- MIN_System_CPU
- MAX_System_CPU
- AVG_System_CPU
- SUM_System_CPU

- **COUNT**
- Attributes that have increasing numeric values with occasional resets (for example, counts of x since ….). These attributes are aggregated with TOTAL, HIGH, LOW, and LATEST values from the detailed data to the appropriate summarization period. There will be four attributes in the summarized table using the original attribute name prefixed with TOT_, HI_, LOW_ , and LAT_. For example the "Linux_System_Statistics_H" table would have the following attributes for the System_Uptime attribute:

  - TOT_System_Uptime
  - HI_System_Uptime
  - LOW_System_Uptime
  - LAT_System_Uptime

  Count type attributes use delta-based aggregation. Delta-based aggregation algorithms calculate the delta between two intervals and use that number as the stored value. For example, if you have an attribute that is the total amount of cache hits since the system has been started, then a delta-based calculation would compute the difference between each cycle interval. At the end of the summarization period, it would total all deltas, store the high value, store the low value, and store the last value recorded. *IBM Tivoli Monitoring Administrator's Guide*, SC32-9408, has more details about delta-based summarization.

- **PROPERTY**

  Attributes that rarely change (for example, total amount of memory or CPU speed). There will be one attribute in the summarized table using the original attribute name prefixed with just LAT_. For example, the "Linux_VM_Stats_Q" (Memory) table would have the following attribute for the Total_Swap_Space attribute:

  LAT_Total_Swap_Space

- **PEAK**

  Attributes that are high-water marks or snapshot based. There will be one attribute in the summarized table using the original attribute name prefixed with just MAX_. For example the “Linux_Swap_Rate_Y” table would have the following attribute for the Peak_Swap_Space_Used attribute:

MAX_Peak_Swap_Space_Used

- **LOW**

  Attributes that are low-water marks or snapshot based. There will be one attribute in the summarized table using the original attribute name prefixed with just MIN_. For example the “Linux_Swap_Rate_Y” table would have the following attribute for the Low_Free_Memory attribute:

  MIN_Low_Free_Memory

The other three types that are rarely used are:

- **SAMPLECOUNT**

  Attributes that are used to calculate the number of intervals that are sampled to get an average. There will be one attribute in the summarized table using the original attribute name prefixed with just SUM.

- **PDEL**

  Attributes that are deltas precalculated by the application (change over a period of time). These attributes are aggregated with a MIN, MAX, and SUM values.

- **STATE**

  Not used at this time. Generally, an enumeration list of options referring to the condition of a resource (for example, up, down).

Note: If a column name exceeds the RDBMS name length (for example, DB2 is 30 characters), the Tivoli Data Warehouse creates an internal column name and stores the internal name and original attribute name in a table called WAREHOUSEID.

4.2.5 Object definitions

In IBM Tivoli Monitoring 6.1, all attribute groups and attributes are defined as object definitions in files called Object Definition Interchange (ODI) files. Most of the Tivoli Data Warehouse V2.1 schema information can be obtained by knowing how to interpret the ODI files. The ODI files can be found on the TEPS server in the default IBM Tivoli Monitoring 6.1 install directory (for example, c:\IBM\IBM Tivoli Monitoring\CNPS). The format of the ODI files are:

`doc`*`nnn`*

(*nnn* is the product identifier.)

For example:

docknt	The ODI file for Windows OS agent
dockux	The ODI file for UNIX OS agent
docklz	The ODI file for Linux OS agent
dockud	The ODI file for DB2 agent

Example 4-1 shows selected parameters from a Windows docknt ODI file.

Example 4-1 docknt ODI file example

```
*TABLE:  WTSYSTEM
*OBJECT: NT_System
*OCCURS: Single
*OPGRP:  COM, CONF
*NLSID:  KNT0000
*INDEX:  IRAKEY USERNAME OSTYPE VERSION
*INDEX:  IRAKEY NETADDRESS NUMOFPRCSR PRCSSRTYPE PAGESIZE PCTTLPRIVT
*INDEX:  IRAKEY PCTTLPCSRT PCTTLUSERT CTXSWITCH FLECTLBYT FLECTLOP
*INDEX:  IRAKEY FLEDATOP FLEREADBTS FLEREADOP FLEWRTEBTS FLEWRTEOP
*INDEX:  IRAKEY PRCQUELNG SYSCALLSEC SYSUPTIME TTLINTSEC ALIFIXRATE
*INDEX:  IRAKEY EXCDISRATE FLOATERATE
*FILE:   VSAM.WTSYSTEM
*REM:    System Object
*REM:    Index: 002
The System object type includes those counters that apply to all
processors on the computer collectively. These counters represent the
activity of all processors on the computer.

*ATTR:   Processor_Type
*CAPTION:Processor\Type
*COLUMN: PRCSSRTYPE
*TYPE:   I,4
*BEHAV:  PROPERTY
*NLSID:  KNT0020
The type of the processors on the pc.

*ATTR:   %_Total_User_Time
*CAPTION:%_Total\User_Time
*COLUMN: PCTTLUSERT
*TYPE:   I,4
*BEHAV:  GAUGE
*RANGE:  0-100
*ENUM:   Unknown=-1
*NLSID:  KNT0028
The % Total User Time is the average percentage ...

*ATTR:   System_Up_Time
*CAPTION: System Up Time (Seconds)
```

```
*COLUMN: SYSUPTIME
*PRINTF: "%u"
*TYPE:   I,4
*BEHAV:  COUNT
*NLSID:  KNT0050
Total Time (in seconds) that the computer has been operational since
it was last started.
```

The ODI files contain a lot of information describing IBM Tivoli Monitoring 6.1 objects. In this chapter we describe only ODI files for the purposes of finding out information about the Tivoli Data Warehouse V2.1 schema. Therefore we focus on only five ODI keywords:

- ***TABLE**

  Every attribute group is uniquely identified by a *TABLE: keyword in an ODI file. In this example WTSYSTEM identifies the short-term binary file table name for the Windows OS agent. After each *TABLE: keyword are multiple *ATTR: keywords (one for each attribute defined in that table).

- ***OBJECT**

  The *OBJECT: keyword identifies the long-term RDBMS table name. If the table name exceeds the RDBM name limit, an internal name will be used in the RDBMS and the real name will be in the WAREHOUSEID table.

- ***ATTR**

  The *ATTR: keyword identifies the column names used in the RDBMS tables. If the column name exceeds the RDBM name limit, an internal name will be used in the RDBMS and the real name will be in the WAREHOUSEID table.

- ***TYPE**

  The *TYPE: keyword can be used to determine how the column will be stored in the RDBMS. It does not describe the actual DDL but it can give an idea of whether the column is, for example, a string or integer.

- ***BEHAV**

  The *BEHAV: keyword describes the aggregation alogorithm used for summarization and pruning. See “Behavior characterization types” on page 205.

4.3 Planning

In this section we discuss the physical and logical planning considerations for implementing the Tivoli Data Warehouse V2.1. Before reading this section, you should review the Chapter 2, “Architecture and planning” on page 15. The following topics are discussed in this section:

- Physical configuration considerations
- “Logical configuration considerations” on page 214
- “Considerations when implementing the Historical Data Gathering component” on page 224

4.3.1 Physical configuration considerations

When planning the physical layout for the new Data Warehouse components shipped with IBM Tivoli Monitoring 6.1 (Tivoli Data Warehouse V2.1), you have to consider the size of the enterprise. Chapter 2, “Architecture and planning” on page 15 discusses three main categories of enterprises:

- Small to medium enterprises (up to 400 agents)
- Large enterprises (up to 4000 agents)
- Huge enterprises (greater than 4000 agents)

When planning a Tivoli Data Warehouse V2.1 architecture, the three main components to consider when implementing the physical layout of the enterprise are:

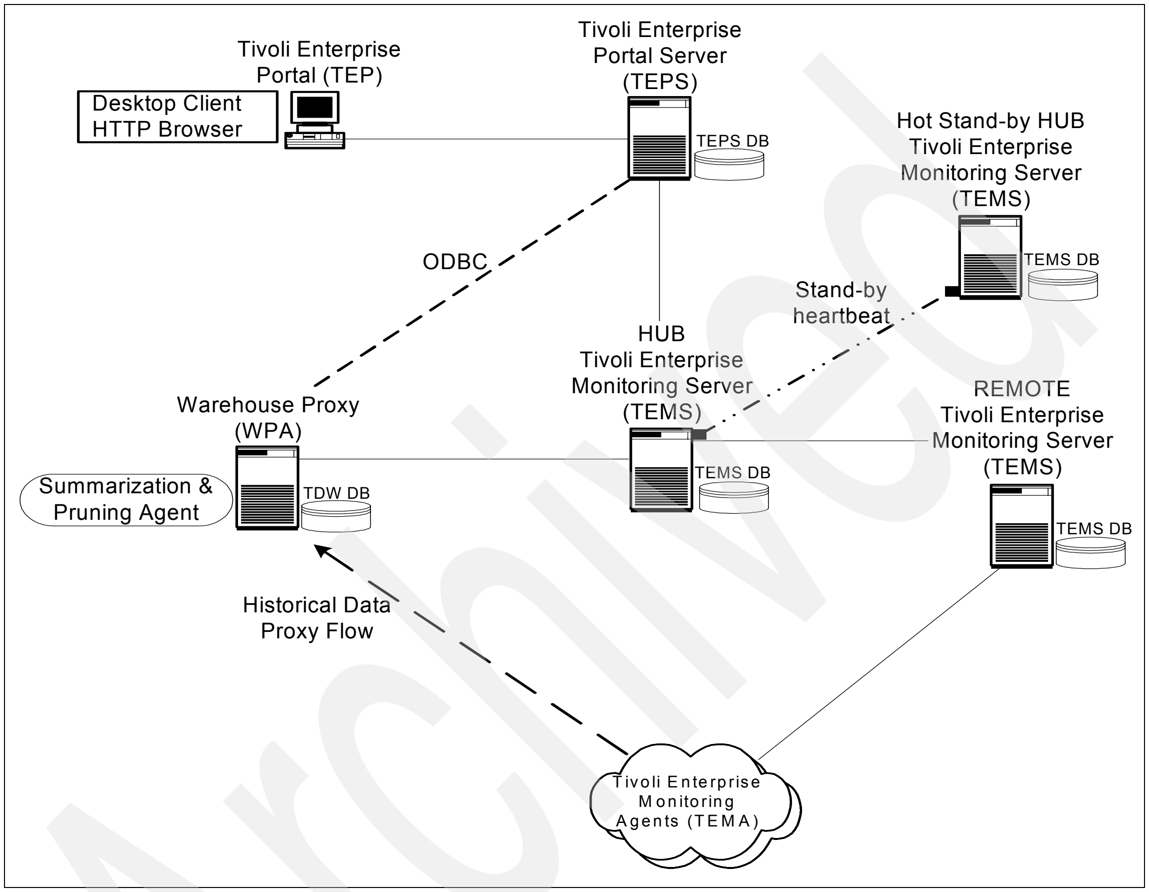
- Warehouse Proxy agent
- Summarization and Pruning agent
- Tivoli Data Warehouse Database Server

Note: Remember when planning the Tivoli Data Warehouse architecture that you can run only one Warehouse Proxy agent per Hub TEMS configuration and only one Summarization and Pruning agent for the entire enterprise (multi-Hub TEMS configurations).

Small to medium enterprise

In a small to medium enterprise it is acceptable to run a minimal Tivoli Data Warehouse configuration. A minimal configuration could have the Warehouse Proxy agent, Summarization and Pruning agent, and the Tivoli Data Warehouse database server on the same machine. The primary caveat to this is that in IBM Tivoli Monitoring 6.1 the Warehouse Proxy agent has to be on a Windows box. Therefore, if an enterprise wanted to run the Warehouse Proxy agent and database server on the same machine it would have to run on a Windows OS. An alternative configuration could be to install the Tivoli Data Warehouse database server (the RDBMS server) on another platform (such as Linux or UNIX). If the database server is installed on a machine other than the Warehouse Proxy agent machine it is also a good idea to install the Summarization and Pruning agent on that same database server machine.

Figure 4-7 is an example of a minimal configuration where the Warehouse Proxy agent, Summarization and Pruning agent, and the Tivoli Data Warehouse database server are all on the same machine.



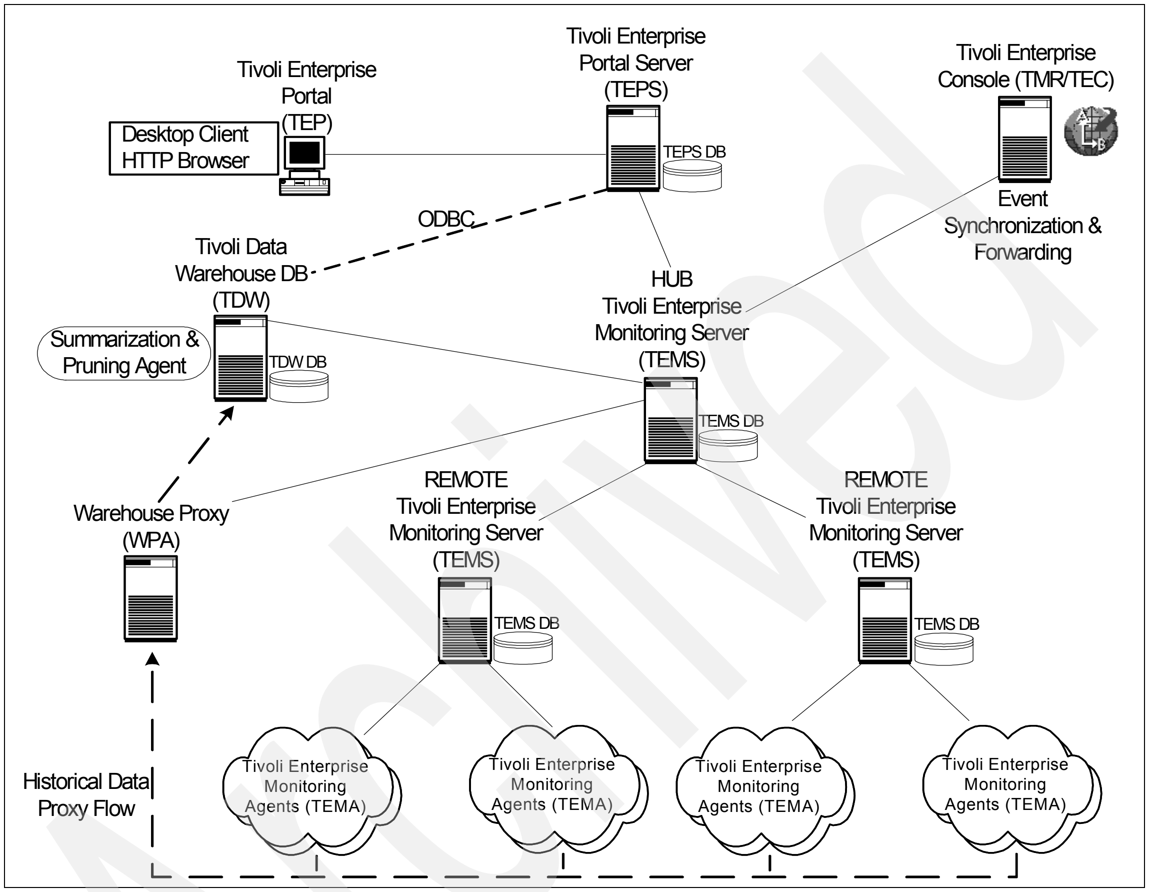
Tivoli Enterprise Portal (TEP)
Desktop Client HTTP Browser
Tivoli Enterprise Portal Server (TEPS)
TEPS DB
Hot Stand-by HUB Tivoli Enterprise Monitoring Server (TEMS)
TEMS DB
ODBC
Stand-by heartbeat
HUB Tivoli Enterprise Monitoring Server (TEMS)
TEMS DB
REMOTE Tivoli Enterprise Monitoring Server (TEMS)
TEMS DB
Warehouse Proxy (WPA)
Summarization & Pruning Agent
TDW DB
Historical Data Proxy Flow
Tivoli Enterprise Monitoring Agents (TEMA)

Figure 4-7 Small to medium topology

Large enterprise

In a large enterprise, the Warehouse Proxy agent should be separated from the Summarization and Pruning agent and Tivoli Data Warehouse database server. In a large enterprise topology it is best to run the Tivoli Data Warehouse database server on its own dedicated machine. It is also recommended that you run the Summarization and Pruning agent on the same machine as the Tivoli Data Warehouse database server.

Figure 4-8 is an example of a large enterprise configuration in which the Warehouse Proxy agent is on one machine and the Summarization and Pruning agent and the Tivoli Data Warehouse database server are on another machine.


Tivoli Enterprise Portal (TEP)
Desktop Client HTTP Browser
Tivoli Enterprise Portal Server (TEPS)
TEPS DB
Tivoli Enterprise Console (TMR/TEC)
Event Synchronization & Forwarding
ODBC
Tivoli Data Warehouse DB (TDW)
Summarization & Pruning Agent
TDW DB
HUB Tivoli Enterprise Monitoring Server (TEMS)
TEMS DB
Warehouse Proxy (WPA)
REMOTE Tivoli Enterprise Monitoring Server (TEMS)
TEMS DB
REMOTE Tivoli Enterprise Monitoring Server (TEMS)
TEMS DB
Historical Data Proxy Flow
Tivoli Enterprise Monitoring Agents (TEMA)
Tivoli Enterprise Monitoring Agents (TEMA)
Tivoli Enterprise Monitoring Agents (TEMA)
Tivoli Enterprise Monitoring Agents (TEMA)
Figure 4-8 Large enterprise topology

Huge enterprise

In a huge enterprise, you must run multiple Hub TEMS due to the limitations of agents supported within a Hub TEMS. Therefore, in a multiple-Hub TEMS topology you can run multiple Warehouse Proxy agent servers. Each Hub TEMS will have its own Warehouse Proxy agent. All of the recommendations described in the large enterprise topology apply for a huge enterprise within each Hub TEMS. The primary consideration in the huge enterprise is that you can only have one Summarization and Pruning agent in the entire enterprise. You should not install the Summarization and Pruning agent in each Hub TEMS. Only one Hub TEMS should have a Summarization and Pruning agent installed. Therefore, the Summarization and Pruning agent must be installed in one specific Hub TEMS instance. This Hub TEMS will act like a logical master. There is no special configuration option for a master Hub TEMS—it is purely a logical consideration in regard to running the Summarization and Pruning agent.

Figure 4-9 is an example of a large enterprise configuration where the Warehouse Proxy agent is on one machine and the Summarization and Pruning agent and the Tivoli Data Warehouse database server are on another machine. Note the dotted circle around the Summarization and Pruning agent: This represents the one logical master Summarization and Pruning agent associated with the one Hub TEMS.

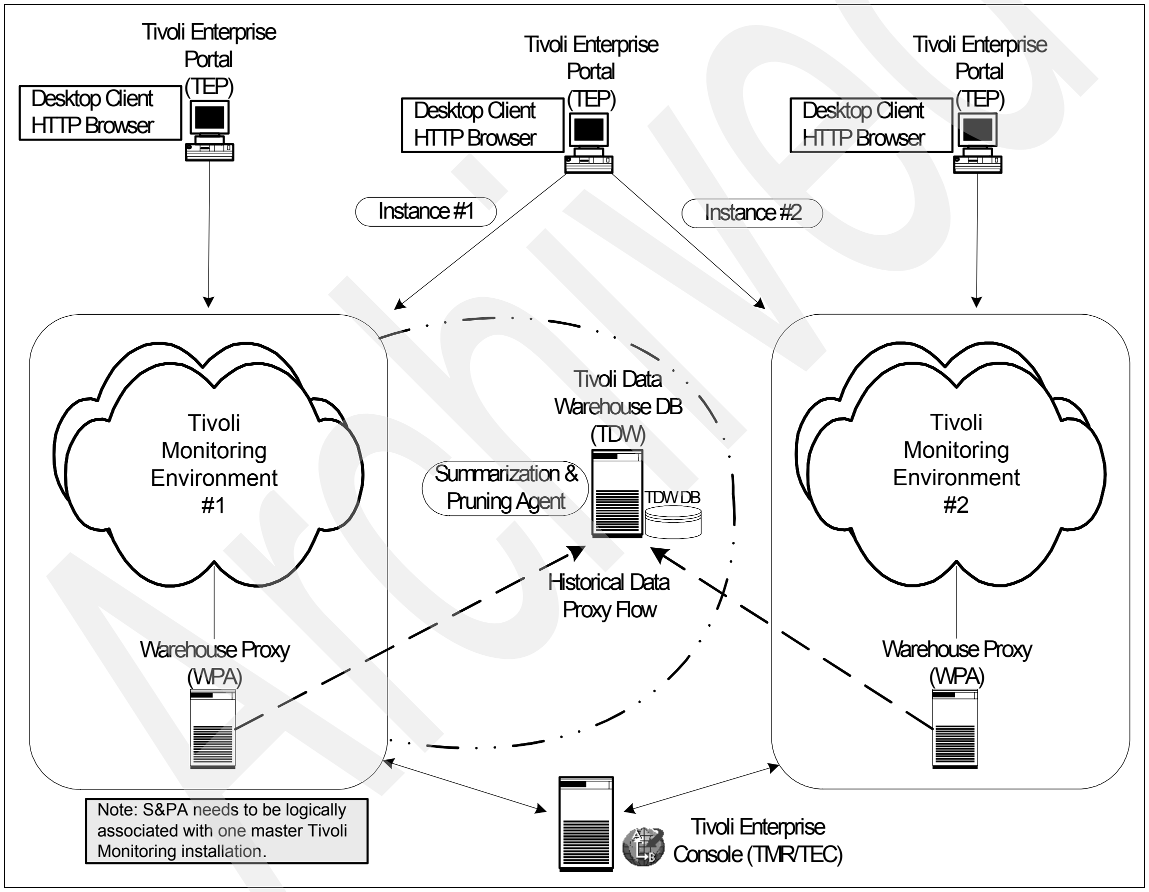

Tivoli Enterprise Portal (TEP)
Desktop Client HTTP Browser
Tivoli Enterprise Portal (TEP)
Desktop Client HTTP Browser
Tivoli Enterprise Portal (TEP)
Desktop Client HTTP Browser
Instance #1
Instance #2
Tivoli Monitoring Environment #1
Tivoli Data Warehouse DB (TDW)
Summarization & Pruning Agent
TDW DB
Tivoli Monitoring Environment #2
Historical Data Proxy Flow
Warehouse Proxy (WPA)
Warehouse Proxy (WPA)
Note: S&PA needs to be logically associated with one master Tivoli Monitoring installation.
Tivoli Enterprise Console (TMR/TEC)

Figure 4-9 Huge enterprise topology

Hardware considerations for the physical layout

The following are hardware considerations for the physical layout:

- Warehouse Proxy agent server
  - Processors: Minimum of four for a large or huge enterprise
- Database server
  - Processors: Minimum of four processors
  - Disk drives: 16 to 24 disk drives for a large or huge enterprise
- Summarization and Pruning agent server
  - Processors: Minimum of four processors
  - Memory: Minimum 2 GB
- Database server and Summarization and Pruning agent on the same machine
  - Processors: Minimum of four processors
  - Memory: Minimum 4 GB
  - Disk drives: 16 to 24 disk drives for a large or huge enterprise

4.3.2 Logical configuration considerations

The primary considerations when planning the logical configuration and implementation of the Tivoli Data Warehouse V2.1 are the estimation of the database size and planning for the kind of data you want to keep in the Tivoli Data Warehouse V2.1 and how long you want to keep it (pruning).

Database sizing

The Tivoli Data Warehouse database has two primary types of tables: detailed and summarization. The *detailed* tables keep raw data. The *summarized* tables keep aggregate data records (such as min, max, and total). The size of the Tivoli Data Warehouse can be estimated based on the total size of all of the detailed tables and all of the summarization tables. When considering table sizes, you also need to know how the data is stored in the tables. In IBM Tivoli Monitoring 6.1, all attribute groups relate to tables in the Tivoli Data Warehouse. Some are multiple instance tables, which have more than one instance collected per interval (for example, process). Multiple instance attribute tables have to be taken into consideration when estimating the size of the database.

Figure 4-4 on page 215 shows the default pruning values for the detailed and summarized tables.

Table 4-4 Default pruning values for detail and summarized tables

Table	Prune timespan
Detail	7 days
Hourly	90 days
Daily	12 months
Weekly	2 years
Monthly	2 years
Quaterly	3 years
Yearly	3 years

Detailed table size calculations

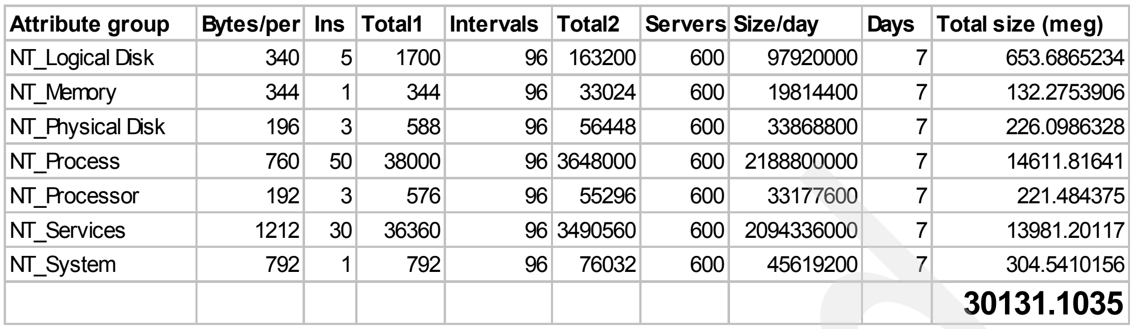
Perform the following steps for the detailed tables:

1. Get a list of all of the attribute groups you plan to collect.
2. Get the record size of each table that corresponds to each attribute group. The detailed record sizes are documented in the Agent user guides (shipped with the product). However, it is always more accurate to use an RDBMS tool (such as DB2CC).
3. Calculate the number of instances for each attribute group you are collecting. This number will most likely be an estimate. For example, processes might be 50, file systems might be 6.
4. Determine the number of intervals per day each attribute group will be collected. This is based on the collection interval. For example, 15-minute intervals would be 96 intervals per day.
5. Determine the number of machines that each attribute group will collect historical data on.
6. Determine for each attribute group the number of days to keep detailed data.

The algorithm is:

```
(size of record *
number of instances *
number of intervals per day *
number of servers *
total days kept = total size of all detailed tables)
```

If we take a simple example of collecting seven Windows OS attribute groups for 600 servers and keep seven days of detailed data, we will have 30131 MB (29.4 GB) allocated for the Windows OS detailed tables in the Tivoli Data Warehouse. This is shown in Figure 4-10 on page 216.



Attribute group	Bytes/per	Ins	Total1	Intervals	Total2	Servers	Size/day	Days	Total size (meg)
NT_Logical Disk	340	5	1700	96	163200	600	97920000	7	653.6865234
NT_Memory	344	1	344	96	33024	600	19814400	7	132.2753906
NT_Physical Disk	196	3	588	96	56448	600	33868800	7	226.0986328
NT_Process	760	50	38000	96	3648000	600	2188800000	7	14611.81641
NT_Processor	192	3	576	96	55296	600	33177600	7	221.484375
NT_Services	1212	30	36360	96	3490560	600	2094336000	7	13981.20117
NT_System	792	1	792	96	76032	600	45619200	7	304.5410156
									30131.1035

Figure 4-10 Windows OS detailed tables database sizing example

The values in the table are:

- Attribute group

  This is the name of the attribute group that is collecting historical data. Each attribute group creates a unique table with the same name stored in the Tivoli Data Warehouse.

- Bytes/per

  This is the record length of the rows in the table. This value can be found in the IBM Tivoli Monitoring 6.1 Agent user guides (online version shipped with the product) or from an RDBMS tool (such as DB2CC).

- Ins

  This is the estimated amount of instances for each monitoring cycle. For example, for NT_Process every interval will create about 50 records, which would allocate 38000 bytes (760 * 50) per interval. In the same example we have listed five logical disks for NT_LogicalDisk. In UNIX, the Disk table might be much higher (one for each file system). However, by far the process and services type tables are going to consume the largest amount of disk space.

- Total1

  This is the bytes per record * total number of estimated instances for that attribute group.

- Intervals

  The total number of intervals per day. In this example all attribute groups were defined to collect every 15 minutes (96 intervals per day). Obviously the collection interval of different attribute groups can greatly effect the overall size of the Tivoli Data Warehouse database. Some attribute groups can be collected at lower intervals (for example, NT_Processor at five minutes).

- Total2

  This is a continuation of the previous calculation. For example, for NT_Process it is (96 * 38000 = 3658000 bytes).
- Servers

  Total number of servers from which this attribute group will be collected.
- Size/day

  Total size per day for a specific attribute group. For NT_Process it is (3648000 * 600 = 2188800000 bytes).
- Days

  Total number of days to keep detailed data for this specific attribute group. The total amount of data is determined by the pruning settings for the attribute group.
- Total size

  This is the total size estimate for a specific attribute group.

For the example in Figure 4-10 on page 216, the total estimated size for the database is 30131 MB (29.4 GB). This example uses only the seven attribute groups for Windows OS. Similar calculations have to be made for other platforms and attribute groups. The total size also varies depending on additional size for DB indexes, log space, free space, and other database space required.

Summarized table size calculations

To estimate the summarized tables, you should perform the following steps:

1. Get a list of all of the attribute groups you plan to collect.
2. Get the record size of each summarized table that corresponds to each attribute group. All of the summarized period tables will have the same length (hourly _H, daily _D, weekly _W, monthly _M, quarterly _Q, or yearly _Y). As this book was being written, the record lengths for the summarized tables were not published. The record sizes can be determined by using a database tool provided with the RDBMS product. For example for this exercise the DB2 Control Center “Estimate Size” option was used.
3. Calculate the number of instances for each attribute group you are collecting. This number will most likely be an estimate; for example, processes might be 50, file systems might be 6. If the estimation for the detailed table was 50 then the summarized table should use the same estimation for the number of instances. In this example we can look at the “NT_Process_D” table and it will have the same number of estimated instances as the detailed table.
4. Determine the total number of summarized records that will be kept. This calculation is based on how long the data will be retained for each

summarization period (hourly, daily, weekly, monthly, quaterly, or yearly). Based on the recommended values specified in Figure 4-4 on page 215, the total number of summarized records would be 2668. The following list shows how this number is derived:

90 days of hourly data (24*90=2160)

12 months of daily data (365/12*12=365)

2 years of weekly data (2*52=104)

2 years of monthly data (2*12=24)

3 years of quarterly data (3*4=12)

3 years of yearly data (3*1=3)

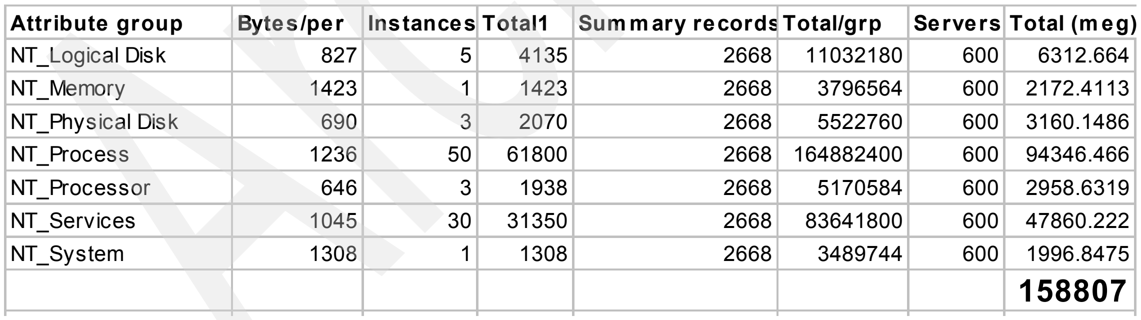
Total = 2668 summarized records

5. Determine the number of machines that each attribute group will collect historical data on.

The algorithm is as follows:

```
(size of record *
number of instances *
number of summarized records total *
number of servers = total size of all summarized tables)
```

If we use the same seven tables used in Figure 4-10 on page 216 and also use Figure 4-4 on page 215 for the default summarization pruning values for the same 600 servers, we come up with a total of 158807 MB (155 GB) for all of the summarization tables, as shown in Figure 4-11.



Attribute group	Bytes/per	Instances	Total1	Summary records	Total/grp	Servers	Total (meg)
NT_Logical Disk	827	5	4135	2668	11032180	600	6312.664
NT_Memory	1423	1	1423	2668	3796564	600	2172.4113
NT_Physical Disk	690	3	2070	2668	5522760	600	3160.1486
NT_Process	1236	50	61800	2668	164882400	600	94346.466
NT_Processor	646	3	1938	2668	5170584	600	2958.6319
NT_Services	1045	30	31350	2668	83641800	600	47860.222
NT_System	1308	1	1308	2668	3489744	600	1996.8475
							158807

Figure 4-11 Windows OS summary tables database sizing example

The values in the table are as follows:

- Attribute group

  This is the name of the attribute group that is collecting historical data. Each attribute group will create multiple summarization tables in the Tivoli Data

Warehouse. There will be a unique table for each time period configured. For example, if the NT_Processor attribute group is turned on for hourly and daily summarization, there will be an NT_Processor_H and a NT_Procesor_D table created in the Tivoli Data Warehouse.

- Bytes/per

  This is the record length of the summarized table. All summarized tables of the same type have the same length. For example, NT_Processor_H and NT_Processor_D will have the same record lengths. The summarized table lengths are not documented and should be determined from an appropriate RDBMS tool.

- Instances

  This is the estimated amount of instances for each monitoring cycle. For example, for NT_Process every interval will create 50 records, which would allocate 61800 bytes (12360 * 50) per summary record. In UNIX the Disk table might have 10 or 15 instances (one for each file system). However, by far the process and services type tables will consume the most disk space.

- Total1

  This is instances * bytes per record

- Summary records

  This is the total number of estimated summary records for the specific attribute group. The calculation for this is in list item 4 on page 217.

- Total/grp

  This is the total number of bytes per system for each attribute group.

- Servers

  Total number of servers from which this attribute group will be collected.

- Total size

  This is the total size estimate for a specific attribute group for all summary records.

Note: In this example we have turned off Shift Zone in the Summarization and Pruning configuration. Shift Zones ON is the default setting. If the Shift Zones are turned on there would be two extra records for each summarization period. If the default Shift records were left on we would have a total of 8004 summary records, and the total space required for summary records would be 465 GB in this example.

For our simple example of 600 servers and seven Windows OS attribute groups, we reach a total of 29 GB for the detailed tables and 155 GB for the summary tables for a grand total of 185 GB. Total space calculations depend on many factors including but not limited to DB indexes, log space, free space, and other database space.

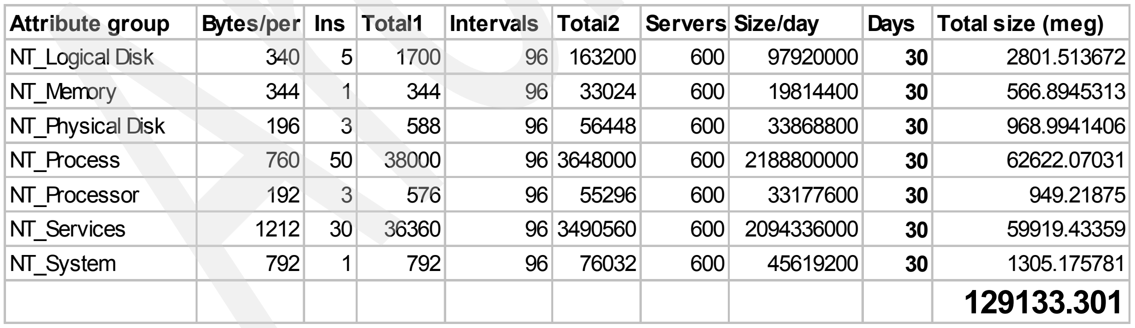
Tuning the size of your Tivoli Data Warehouse database

Four main factors can affect the size of the Tivoli Data Warehouse V2.1 database:

1. The amount of detailed data you keep in the historical database.
2. The amount of hourly summarized data you keep in the historical database.
3. Using shift data in the historical database.
4. Collecting data for multiple instance attribute groups.

The amount of detailed data you keep in the historical database

The example in Figure 4-10 on page 216 uses seven days of detailed data for 600 Windows servers and the total is 29 GB. If we simply change that value to 30 days of detailed data, the total size for all seven attribute groups becomes 129133 MB or 126 GB (Figure 4-12). The key consideration is that in Tivoli Data Warehouse V1.x, all of the data was aggregated before it left the endpoint. In Tivoli Data Warehouse V2.1 all of the raw data that is collected can be stored in the Tivoli Data Warehouse and the amount of days that detailed data is maintained will greatly affect the size of the database. Another factor that affects the size of the detailed tables in the Tivoli Data Warehouse is the collection interval. The examples in this section all use 15-minute intervals (96 per day). However, increasing or lowering the collection intervals on different attribute groups will have an obvious effect on the total size required in the Tivoli Data Warehouse. Therefore, setting something like NT_Memory to 5 minutes might be acceptable. However, NT_Process less than 15 minutes could require a lot of disk space.



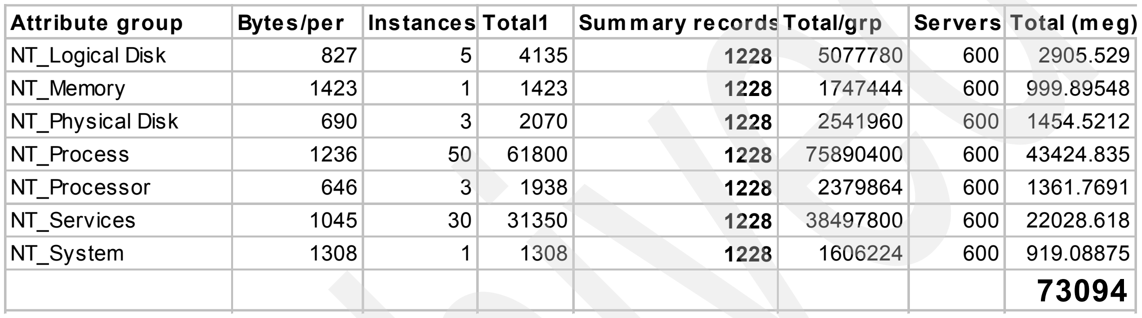
Attribute group	Bytes/per	Ins	Total1	Intervals	Total2	Servers	Size/day	Days	Total size (meg)
NT_Logical Disk	340	5	1700	96	163200	600	97920000	**30**	2801.513672
NT_Memory	344	1	344	96	33024	600	19814400	**30**	566.8945313
NT_Physical Disk	196	3	588	96	56448	600	33868800	**30**	968.9941406
NT_Process	760	50	38000	96	3648000	600	2188800000	**30**	62622.07031
NT_Processor	192	3	576	96	55296	600	33177600	**30**	949.21875
NT_Services	1212	30	36360	96	3490560	600	2094336000	**30**	59919.43359
NT_System	792	1	792	96	76032	600	45619200	**30**	1305.175781
									129133.301

Figure 4-12 Detailed data kept for 30 days

Hourly summarized data you keep in the historical database

In the examples used in Figure 4-10 on page 216 and Figure 4-11 on page 218 the summarized data space is more than five times as large as the detailed data space required (155 GB versus 29 GB). If we change the hourly summarized data to 30 days instead of 90 days, we reduce the size of the database by half. Figure 4-13 shows an example of changing the default 90 days hourly to 30 days. This is reflected in the total number of summary records. For example:

- 30 days of hourly data (24*30=720)
- 12 months of daily data (365/12*12=365)
- 2 years of weekly data (2*52=104)
- 2 years of monthly data (2*12=24)
- 3 years of quarterly data (3*4=12)
- 3 years of yearly data (3*1=3)
- Total = 1228 summarized records



Attribute group	Bytes/per	Instances	Total1	Summary records	Total/grp	Servers	Total (meg)
NT_Logical Disk	827	5	4135	**1228**	5077780	600	2905.529
NT_Memory	1423	1	1423	**1228**	1747444	600	999.89548
NT_Physical Disk	690	3	2070	**1228**	2541960	600	1454.5212
NT_Process	1236	50	61800	**1228**	75890400	600	43424.835
NT_Processor	646	3	1938	**1228**	2379864	600	1361.7691
NT_Services	1045	30	31350	**1228**	38497800	600	22028.618
NT_System	1308	1	1308	**1228**	1606224	600	919.08875
							73094

Figure 4-13 Summarized tables with 30 days of hourly data

Using shift data in the historical database

The default configuration for summarization and pruning is to have shift data stored. The examples in this chapter assume that shift data is not enabled. However, if shift data is enabled there will be two additional records in all of the collected summarized tables in the Tivoli Data Warehouse. Therefore, in the original example in Figure 4-11 on page 218, the total space required would be three times the original 155 GB example (465 GB). The key consideration for shift data being collected is the need for collecting shift data. Are there three SLA/SLOs in your organization that require reports and data analysis to be done on a shift basis? If the answer is yes, then it is a fantastic feature. If the answer is no, then you can cut your disk space requirements by more than half.

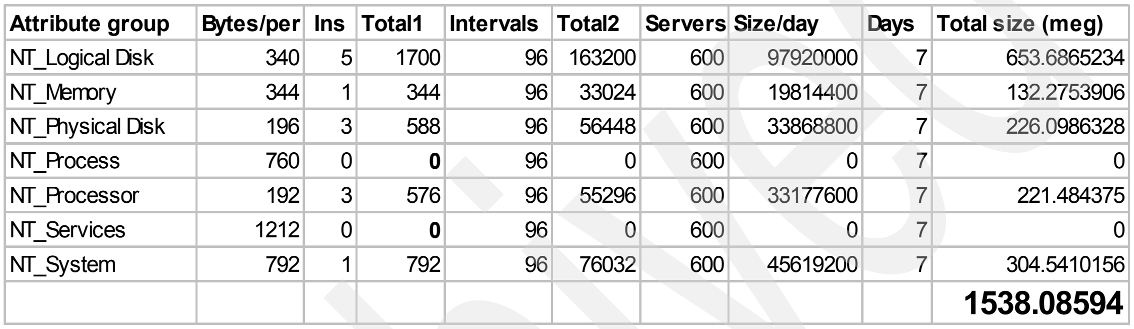
Collecting data for multiple instance attribute groups

A key consideration when analyzing database size is determining what type of attribute groups are collected. Attribute groups that collect a large amount of instances are good candidates to consider for database size tuning. The process and service-related tables are a good place to start. In IBM Tivoli Monitoring 5.x and Tivoli Data Warehouse V1.x process not all process instances were collected into the Tivoli Data Warehouse. For example, in UNIX only processes that exceeds a CPU threshold were logged to the Tivoli Data Warehouse (and they were also aggregated). Now in Tivoli Data Warehouse V2.1, all processes that are running on a system will be collected to the Tivoli Data Warehouse. If we use the example in Figure 4-10 on page 216 and Figure 4-11 on page 218 and we turn off NT_Process and NT_services for historical collection, the numbers are reduced from a total of 184 GB to a total of 17 GB. See the totals in Figure 4-14 and Figure 4-15.

The calculation is as follows:

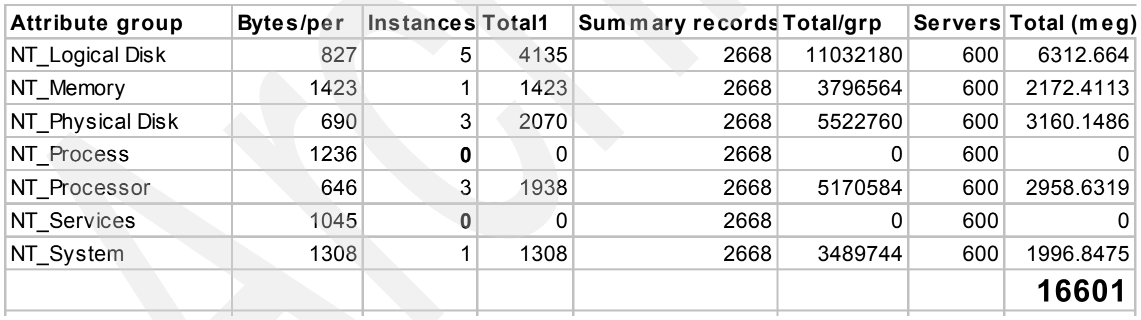
1538 MB detail + 16601 MB summary = 18139 MB total

18139/1024 = 17.7 GB



Attribute group	Bytes/per	Ins	Total1	Intervals	Total2	Servers	Size/day	Days	Total size (meg)
NT_Logical Disk	340	5	1700	96	163200	600	97920000	7	653.6865234
NT_Memory	344	1	344	96	33024	600	19814400	7	132.2753906
NT_Physical Disk	196	3	588	96	56448	600	33868800	7	226.0986328
NT_Process	760	0	**0**	96	0	600	0	7	0
NT_Processor	192	3	576	96	55296	600	33177600	7	221.484375
NT_Services	1212	0	**0**	96	0	600	0	7	0
NT_System	792	1	792	96	76032	600	45619200	7	304.5410156
									1538.08594

Figure 4-14 Detailed tables with NT_Process and NT_Services turned off

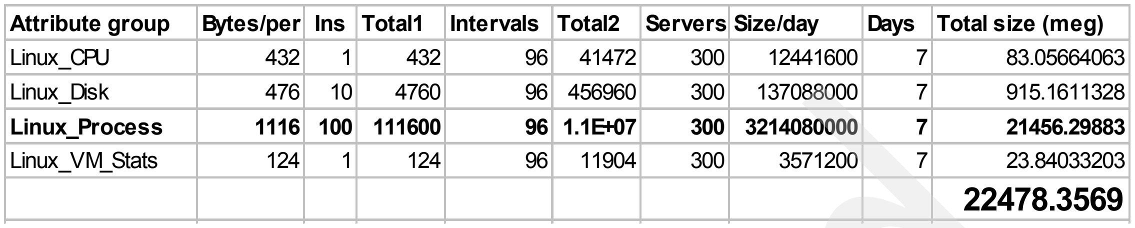


Attribute group	Bytes/per	Instances	Total1	Summary records	Total/grp	Servers	Total (meg)
NT_Logical Disk	827	5	4135	2668	11032180	600	6312.664
NT_Memory	1423	1	1423	2668	3796564	600	2172.4113
NT_Physical Disk	690	3	2070	2668	5522760	600	3160.1486
NT_Process	1236	**0**	0	2668	0	600	0
NT_Processor	646	3	1938	2668	5170584	600	2958.6319
NT_Services	1045	**0**	0	2668	0	600	0
NT_System	1308	1	1308	2668	3489744	600	1996.8475
							16601

Figure 4-15 Summarized tables with NT_Process and NT_Services turn off

An alternative to completely turning off the NT_Process and NT_Services attribute groups is to lower the amount of days kept for those groups. For example, maybe only keep one day of detailed data and seven days of hourly data for those two groups.

All of the examples listed above were based on the Windows OS attribute groups. However, similar calculations can be performed for UNIX and Linux attribute groups. Figure 4-16 on page 223 is an example of using some Linux OS attribute groups. Her again, we see that the process group (Linux_Process) takes up almost 95 percent of the space required for Linux OS historical collections.



Attribute group	Bytes/per	Ins	Total1	Intervals	Total2	Servers	Size/day	Days	Total size (meg)
Linux_CPU	432	1	432	96	41472	300	12441600	7	83.05664063
Linux_Disk	476	10	4760	96	456960	300	137088000	7	915.1611328
Linux_Process	**1116**	**100**	**111600**	**96**	**1.1E+07**	**300**	**3214080000**	**7**	**21456.29883**
Linux_VM_Stats	124	1	124	96	11904	300	3571200	7	23.84033203
									22478.3569

Figure 4-16 Sample Linux OS monitoring attribute groups

Table 4-5 lists other multiple-instance attribute groups to look out for.

Table 4-5 Attribute groups that can have a large number of instances

Table	Product	Estimated instances
File_Information	UNIX OS or Linux OS	100
Process	UNIX OS	100-1000
Linux_Process	Linux OS	100-1000
Linux_Socket_Status	Linux OS	10-100
Linux_Socket_Detail	Linux OS	10-100
NT_Process	Windows OS	50-100
NT_Services	Windows OS	30-100
NT_Thread	Windows OS	100-1000

Final considerations

After researching the amount of data that is being summarized and pruned over a period of days, you should run a few servers and agents for a few days to validate your estimates. Remember, the calculations in this section do not take into account additional size for DB indexes, log space, and free space. After running the servers and agents for a few days you can adjust your estimates accordingly.

4.3.3 Considerations when implementing the Historical Data Gathering component

When implementing the Historical Data Gathering component, consider:

- There can only be one Warehouse Proxy agent per Hub TEMS and the Warehouse Proxy agent must run on Windows.
- There can be only one Summarization and Pruning agent in an IBM Tivoli Monitoring 6.1 enterprise.
- There is a know limitation using IP.PIPE when collecting historical data from agents. Because the Warehouse Proxy agent can run only on Windows with IBM Tivoli Monitoring 6.1, there is a limit to the amount of IP.PIPE connections that can be made on a Windows system. Therefore, the recommended effective maximum number of agents per Warehouse Proxy agent is around 1500. The IP.UDP does not have this limitation.

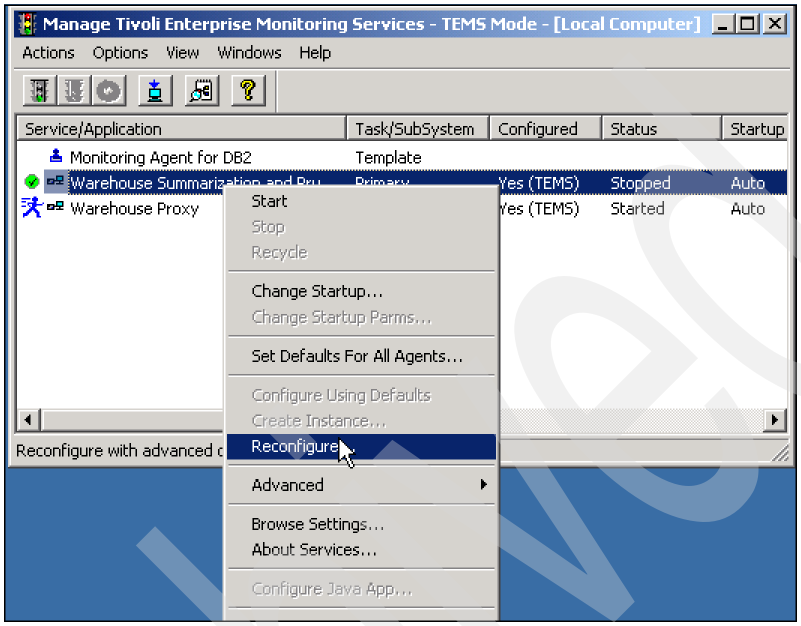
4.4 Configuration

There are two parts to configuring the Tivoli Data Warehouse V2.1 in IBM Tivoli Monitoring 6.1: configuring the Summarization and Pruning agent default parameters and configuring the specific agent attribute groups from the TEP History configuration icon.

4.4.1 Configuring Summarization and Pruning agent

When IBM Tivoli Monitoring 6.1 is installed, the Summarization and Pruning agent can be configured with default values. The default values that are set during the installation of the Summarization and Pruning agent can be used as default values for all of the agent default attribute groups. If the scheduled summarization and pruning process (for example, once per day process) has not run for the first time then the defaults for all agent default attribute groups can be reconfigured. It is recommended that the Summarization and Pruning agent not be started and scheduled to run before the first time defaults are configured. A thorough review of the planning section of this chapter should be completed before configuring the Summarization and Pruning agent default settings. Find more about reconfiguring the Summarization and Pruning agent settings in 3.2.13, “Summarization and Pruning agent installation and configuration” on page 139.

Figure 4-17 shows how to reconfigure the Summarization and Pruning agent settings from the Management Tivoli Enterprise Monitoring Services console (right-click the agent and choose Reconfigure).


Manage Tivoli Enterprise Monitoring Services - TEMS Mode - [Local Computer]

Service/Application	Task/SubSystem	Configured	Status	Startup
Monitoring Agent for DB2	Template			
Warehouse Summarization and Pru...	Primary	Yes (TEMS)	Stopped	Auto
Warehouse Proxy		Yes (TEMS)	Started	Auto

Reconfigure with advanced c

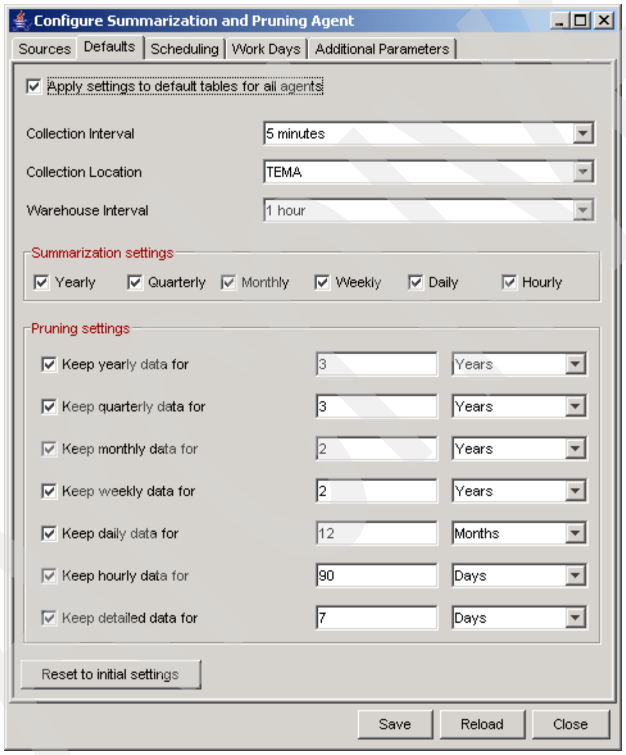
Figure 4-17 Summarization and Pruning agent configuration

Figure 4-18 is an example of the Summarization and Pruning agent default setting panel. If the scheduled summarization and pruning process has never run, the values can be reconfigured and will be used by all of the agent default attribute groups.

Note: If you want to reapply default settings for an agent that has already had summarization and pruning work done to it, delete the ksy.k<pc>.installed file, where <pc> is the two-letter product code for the agent:

On a Windows system: in the install_dir\tmaIBM Tivoli Monitoring6\logs directory

On a UNIX or Linux system: install_dir\logs directory



☑ Apply settings to default tables for all agents

Collection Interval	5 minutes
Collection Location	TEMA
Warehouse Interval	1 hour

Summarization settings

☑ Yearly ☑ Quarterly ☑ Monthly ☑ Weekly ☑ Daily ☑ Hourly

Pruning settings

☑ Keep yearly data for	3	Years
☑ Keep quarterly data for	3	Years
☑ Keep monthly data for	2	Years
☑ Keep weekly data for	2	Years
☑ Keep daily data for	12	Months
☑ Keep hourly data for	90	Days
☑ Keep detailed data for	7	Days

Figure 4-18 Summarization and Pruning agent default configuration panel

These are the fields in Figure 4-18 on page 226.

- Apply settings to default tables for all agents

  If this option is selected, all of the agent default attribute groups will inherit the defaults specified on this screen. After the summarization and pruning scheduled run has completed, changes to this screen will not effect the agent default attribute groups settings.

- Collection interval

  The collection interval sets the default time to collect data in the binary files. The location of the binary files depends on the collection location setting. The default five-minute value might be a little low for all default attribute groups.

- Collection location

  This is the default location for storing the binary files. Whenever possible, select TEMA (at the agent).

- Warehouse interval

  This is the interval that the TEMA or TEMS binary data will be uploaded to the Warehouse Proxy agent. The options are one hour or daily. Environments with a lot of agents are recommended to select one hour.

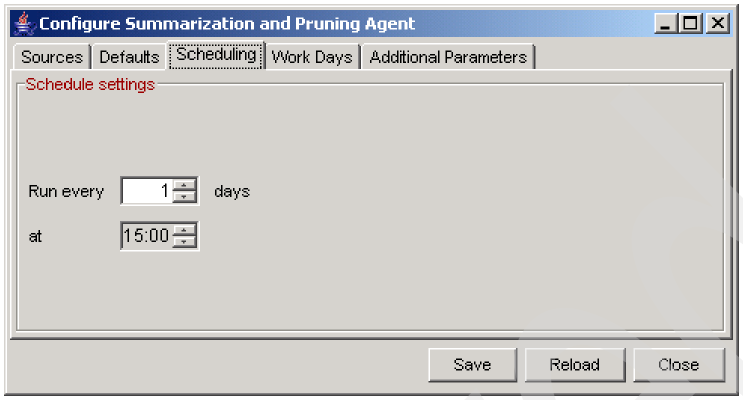
- Summarization settings

  This enables you to select the summarization tables that will be created in the Tivoli Data Warehouse and used for aggregation.

- Pruning settings

  This sets the time to keep data in the Tivoli Data Warehouse. Data older than the prune settings is removed from the Tivoli Data Warehouse.

Figure 4-19 is an example of the Summarization and Pruning agent scheduling panel.



Schedule settings

Run every 1 days

at 15:00

Figure 4-19 Summarization and Pruning agent Scheduling configuration panel

The fields in Figure 4-19 are:

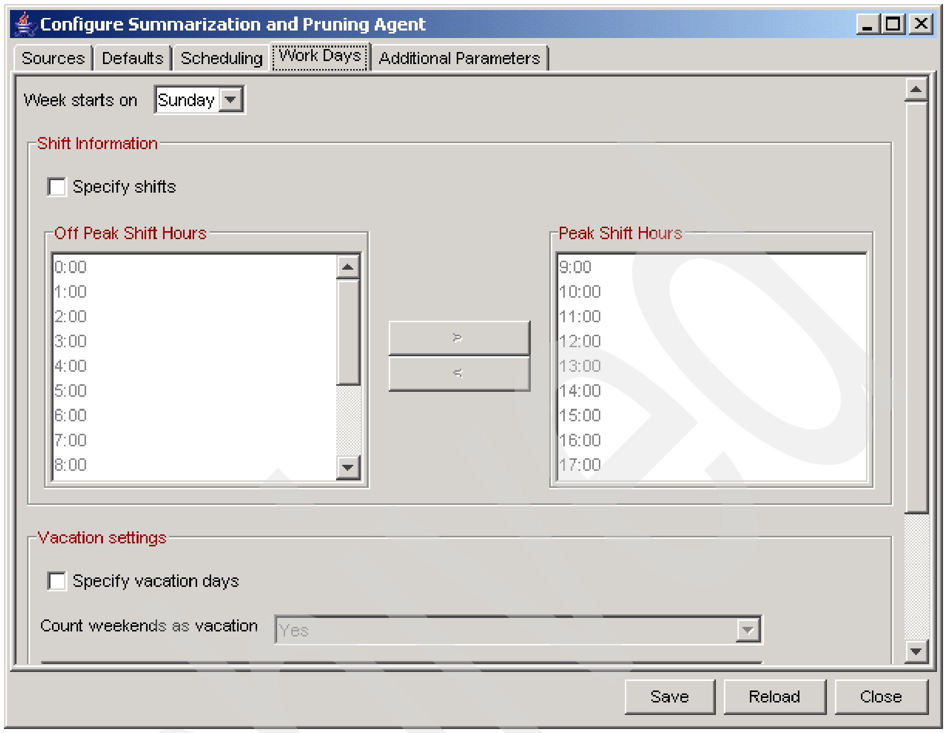
- Run every

  This option sets the daily cycle time. The default is one day. However it could be set to run every seven days. It is recommended that summarization and pruning run happens every day.

- at

  This value is the time the summarization and pruning run is scheduled every day. The default is 02:00 a.m.

Figure 4-20 shows the Summarization and Pruning agent Work Days tab.



Week starts on: Sunday

Shift Information

☐ Specify shifts

Off Peak Shift Hours	Peak Shift Hours
0:00	9:00
1:00	10:00
2:00	11:00
3:00	12:00
4:00	13:00
5:00	14:00
6:00	15:00
7:00	16:00
8:00	17:00

Vacation settings

☐ Specify vacation days

Count weekends as vacation: Yes

Figure 4-20 Summarization and Pruning agent Work Days configuration tab

The Work Days tab includes these fields:

- Week starts on

  If shift data is used, this sets the start day of the week.

- Specify shifts

  This option allows you to set peak and non-peak shifts. If this check box is selected two additional records will be created for each attribute group in the summary tables. Since all data is aggregated (rolled up) from the detail there will be three different summary records for each interval of an instance. For example, the NT_Memory_D will have three records for each day:

  – One summarized record for all hours in the day

  – One summarized record for off-peak hours per day

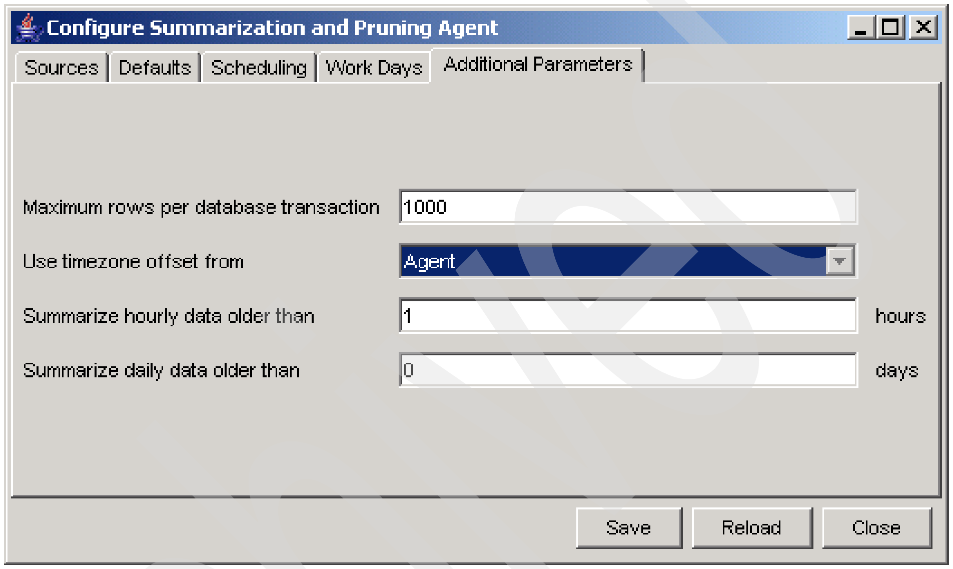
  – One summarized record for peak hours per day

- Specify vacation days

  Additional historical data can be summarized based on vacation day settings.

Note: Changing the shift information after data has been summarized can create an inconsistency in the data. Previous data collected and summarized cannot be recalculated with the new shift values.

Figure 4-21 shows the Summarization and Pruning agent Additional Parameters tab.



Maximum rows per database transaction: 1000
Use timezone offset from: Agent
Summarize hourly data older than: 1 hours
Summarize daily data older than: 0 days

Figure 4-21 Summarization and Pruning agent Additional Parameters

The Additional Parameters tab includes these fields:

- Maximum row per database transaction

  Specify the maximum rows that can be deleted in a single transaction
- Use timezone offset from

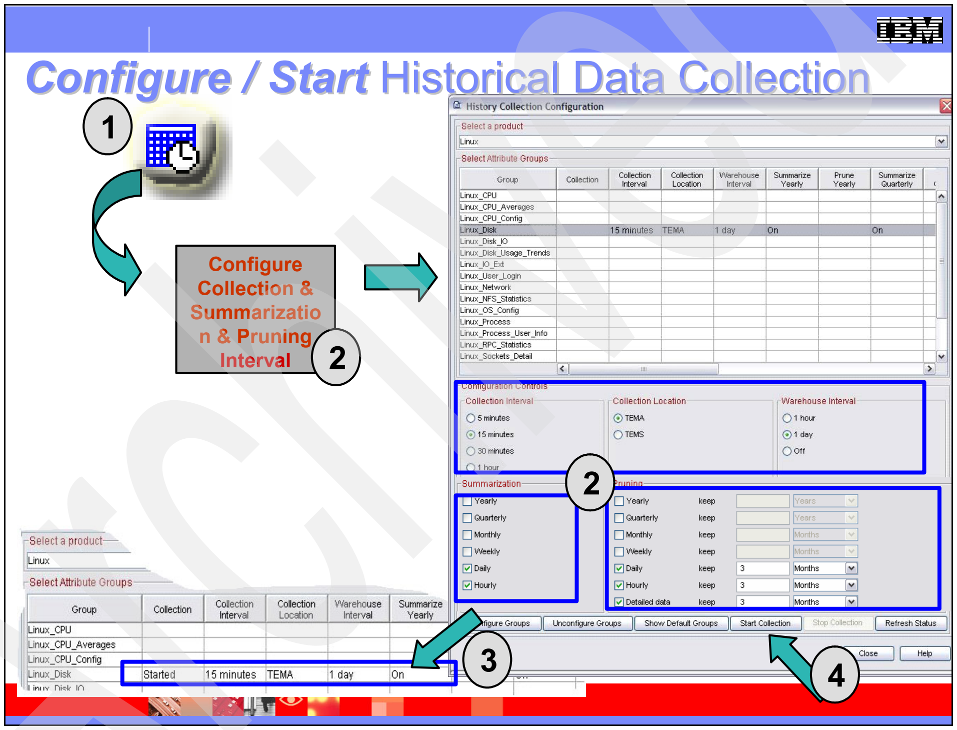
  This is a pull-down list that specifies the source for the timezone that is used. If the Tivoli Data Warehouse servers and agents are not all in the same time zone, and all the data is stored in the same database, use this option to identify the time zone you want to use.
- Summarize hourly data older than, Summarize daily data older than

  Specify the age of the data you want summarized in the Tivoli Data Warehouse. Values are 0 through n. The default is 1 for hourly data and 0 for daily data.

After the default Summarization and Pruning agent configurations are done the Summarization and Pruning agent process should be started. The process wakes up every five minutes to check whether it needs to schedule the summarization and pruning run. When the summarization and pruning process is completed, the defaults will be permanent and the ksy.k<pc>.installed files will be completed in the logs directory.

4.4.2 History configuration

After the first summarization and pruning process has run, you should configure the individual agent attribute groups. The agent attribute groups can be configured from the TEPS History configuration icon as seen in the steps in Figure 4-22.



Configure / Start Historical Data Collection
1
Configure Collection & Summarization & Pruning Interval
2
2
3
4
History Collection Configuration
Configure Groups
Unconfigure Groups
Show Default Groups
Start Collection
Stop Collection
Refresh Status
Close
Help

Select a product: Linux

Select Attribute Groups

Group	Collection	Collection Interval	Collection Location	Warehouse Interval	Summarize Yearly	Prune Yearly	Summarize Quarterly
Linux_CPU							
Linux_CPU_Averages							
Linux_CPU_Config							
Linux_Disk		15 minutes	TEMA	1 day	On		On
Linux_Disk_IO							
Linux_Disk_Usage_Trends							
Linux_IO_Ext							
Linux_User_Login							
Linux_Network							
Linux_NFS_Statistics							
Linux_OS_Config							
Linux_Process							
Linux_Process_User_Info							
Linux_RPC_Statistics							
Linux_Sockets_Detail							

Configuration Controls

Collection Interval: 5 minutes [✓ 15 minutes] 30 minutes 1 hour

Collection Location: [✓ TEMA] TEMS

Warehouse Interval: 1 hour [✓ 1 day] Off

Summarization: Yearly, Quarterly, Monthly, Weekly, [✓ Daily], [✓ Hourly]

Pruning:

Yearly	keep		Years
Quarterly	keep		Years
Monthly	keep		Months
Weekly	keep		Months
✓ Daily	keep	3	Months
✓ Hourly	keep	3	Months
✓ Detailed data	keep	3	Months

Select a product: Linux

Select Attribute Groups

Group	Collection	Collection Interval	Collection Location	Warehouse Interval	Summarize Yearly
Linux_CPU					
Linux_CPU_Averages					
Linux_CPU_Config					
Linux_Disk	Started	15 minutes	TEMA	1 day	On

Figure 4-22 History collection configuration

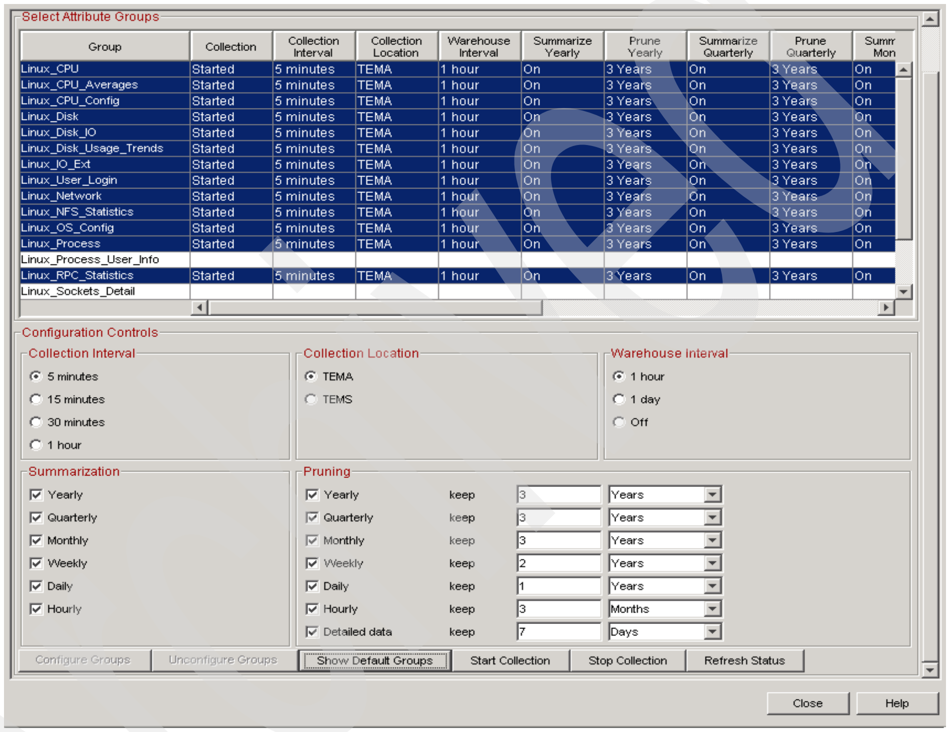
The steps presented in Figure 4-22 are as follows:

1. Select the History configuration icon from the TEPS GUI.
2. Highlight the specific attribute groups you want to collect historical data for and add the configuration settings. If you select the Show Default Groups button, the panel will highlight all of the preconfigured attribute groups for the current agent. This is useful if it is the first time you are setting up an agent for historical collection.

3. Select the **Configure Groups** button for the highlighted groups. If it is the first time you are configuring an attribute group and you have selected Show Default Groups, all of the default settings that were defined in the Summarization and Pruning agent configuration will be loaded.

4. Highlight the specific groups again and select the start button.

Figure 4-23 shows an example of configuring the default attribute groups for the Linux OS agent.



Select Attribute Groups

Group	Collection	Collection Interval	Collection Location	Warehouse Interval	Summarize Yearly	Prune Yearly	Summarize Quarterly	Prune Quarterly	Summ Mon
Linux_CPU	Started	5 minutes	TEMA	1 hour	On	3 Years	On	3 Years	On
Linux_CPU_Averages	Started	5 minutes	TEMA	1 hour	On	3 Years	On	3 Years	On
Linux_CPU_Config	Started	5 minutes	TEMA	1 hour	On	3 Years	On	3 Years	On
Linux_Disk	Started	5 minutes	TEMA	1 hour	On	3 Years	On	3 Years	On
Linux_Disk_IO	Started	5 minutes	TEMA	1 hour	On	3 Years	On	3 Years	On
Linux_Disk_Usage_Trends	Started	5 minutes	TEMA	1 hour	On	3 Years	On	3 Years	On
Linux_IO_Ext	Started	5 minutes	TEMA	1 hour	On	3 Years	On	3 Years	On
Linux_User_Login	Started	5 minutes	TEMA	1 hour	On	3 Years	On	3 Years	On
Linux_Network	Started	5 minutes	TEMA	1 hour	On	3 Years	On	3 Years	On
Linux_NFS_Statistics	Started	5 minutes	TEMA	1 hour	On	3 Years	On	3 Years	On
Linux_OS_Config	Started	5 minutes	TEMA	1 hour	On	3 Years	On	3 Years	On
Linux_Process	Started	5 minutes	TEMA	1 hour	On	3 Years	On	3 Years	On
Linux_Process_User_Info									
Linux_RPC_Statistics	Started	5 minutes	TEMA	1 hour	On	3 Years	On	3 Years	On
Linux_Sockets_Detail									

Configuration Controls

Collection Interval: [✓ 5 minutes] 15 minutes 30 minutes 1 hour

Collection Location: [✓ TEMA] TEMS

Warehouse Interval: [✓ 1 hour] 1 day Off

Summarization: [✓ Yearly] [✓ Quarterly] [✓ Monthly] [✓ Weekly] [✓ Daily] [✓ Hourly]

Pruning

[✓ Yearly]	keep	3	Years
[✓ Quarterly]	keep	3	Years
[✓ Monthly]	keep	3	Years
[✓ Weekly]	keep	2	Years
[✓ Daily]	keep	1	Years
[✓ Hourly]	keep	3	Months
[✓ Detailed data]	keep	7	Days

Figure 4-23 History configuration panel

The fields and buttons in Figure 4-23 include:

- Collection Interval (radio buttons)

  The collection interval sets the default time to collect data on the TEMA or TEMS to the binary files. The default 5 minute value might be a little low for all default attribute groups. This can be configured for one group or a list of highlighted groups.

- Collection Location (radio buttons)

  This is the default location for storing the binary files. It is recommended that whenever possible, select TEMA (at the agent).

- Warehouse Interval (radio buttons)

  This is the interval that the TEMA or TEMS binary data will be uploaded to the Warehouse Proxy agent. The options are 1 hour, daily, or off. It is recommended for environments with a lot of agents to select 1 hour. If the warehouse interval (off) button is selected no data will be collected in the Tivoli Data Warehouse for the selected attribute groups. However, if the attribute group is started with the interval off then the binary data will be collected on the agent; however, it will never be pruned. *IBM Tivoli Monitoring Administrator's Guide,* SC32-9408, has information about pruning the local binary data in this special case.

- Summarization

  These settings specify which summarization tables will be created in the Tivoli Data Warehouse for the specific attribute groups.

- Pruning settings

  This sets how long to keep data in the Tivoli Data Warehouse. Data older than the prune settings will be removed from the Tivoli Data Warehouse.

- Configure Groups (button)

  Click this to configure the highlighted attribute groups' historical configuration settings. You can highlight a single group or multiple groups.

- Unconfigure Groups (button)

  Click this to unconfigure the highlighted attribute groups' historical configuration settings. You can highlight a single group or multiple groups.

- Show Default Groups (button)

  This highlights all of the predefined (by the agent) attribute groups. Click this to configure the highlighted attribute groups' historical configuration settings. You can highlight a single group or multiple groups.

- Start Collection (button)

  Click this to start all of the highlighted attribute groups. You can highlight a single group or multiple groups.

- Stop Collection (button)

  Click this to stop all of the highlighted attribute groups. You can highlight a single group or multiple groups. If one of the highlighted attribute groups is already stopped, this button will be greyed out.

- Refresh Status (button)

  Click this to refresh the status (Started or Stopped) of all the agents.

4.5 Reporting

There are two reporting interfaces you can use to access data in the Tivoli Data Warehouse V2.1. The first reporting interface is through the Tivoli Enterprise Portal (TEP). The TEP can be used to access historical data from any real-time view and to access historical summarized workspaces. The second reporting interface is used to access the data directly from the data warehouse database by using a third-party tool, such as Crystal Reports.

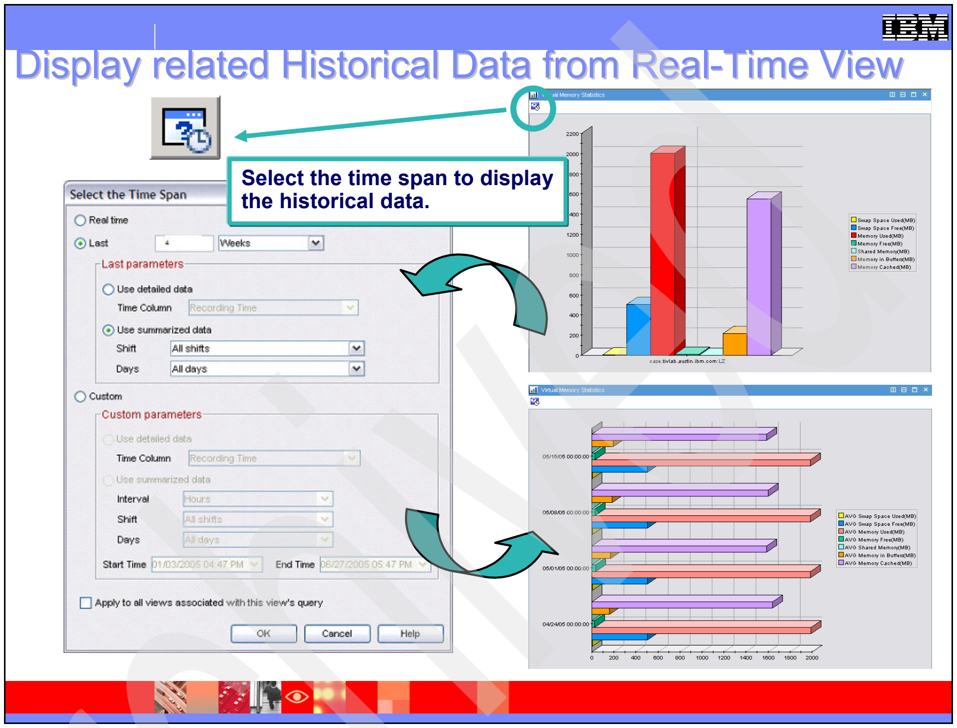
4.5.1 Accessing the data warehouse from the TEP

IBM Tivoli Monitoring 6.1 has seven agents that can collect data into the Tivoli Data Warehouse V2.1:

- Windows
- UNIX
- Linux
- DB2
- MSSQL
- Oracle
- Sybase

After the agents are configured to collect historical data, all of the Tivoli Data reports can be generated from the TEPS.

A report can be viewed from any workspace view by selecting the Timespan icon (Figure 4-24). From the Timespan icon dialog you can change the real-time button to the last button. The last button enables you to specify additional parameters to search detailed or summarized data in the Tivoli Data Warehouse.



Select the Time Span

Real time [✓ Last] 4 Weeks

Last parameters

Use detailed data
Time Column: Recording Time
[✓ Use summarized data]
Shift: All shifts
Days: All days

Custom

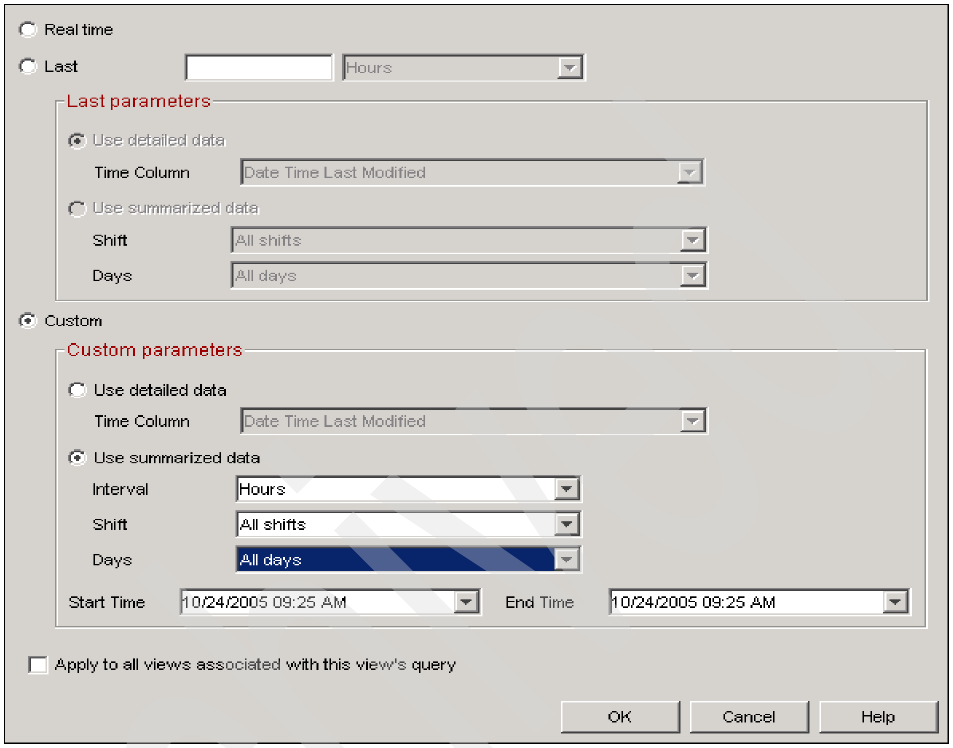
Custom parameters

Use detailed data
Time Column: Recording Time
Use summarized data
Interval: Hours
Shift: All shifts
Days: All days
Start Time: 01/03/2005 04:47 PM End Time: 06/27/2005 05:47 PM

☐ Apply to all views associated with this view's query

Figure 4-24 Using the timespan icon from a real-time view

Figure 4-25 shows an example of how the time span view can be configured to access the Tivoli Data Warehouse.


( ) Real time
( ) Last | | Hours

Last parameters
(●) Use detailed data
Time Column: Date Time Last Modified
( ) Use summarized data
Shift: All shifts
Days: All days

(●) Custom

Custom parameters
( ) Use detailed data
Time Column: Date Time Last Modified
(●) Use summarized data
Interval: Hours
Shift: All shifts
Days: All days
Start Time: 10/24/2005 09:25 AM
End Time: 10/24/2005 09:25 AM

☐ Apply to all views associated with this view's query

Figure 4-25 Configuring the time span for historical reporting

The fields used in Figure 4-25 include:

- Real time

  This is the default selection that instructs the reporting interface to get data from real-time attributes.

- Last

  If this radio button is selected, the report interface retrieves data from either the binary file tables on the TEMA or TEMS or directly from the data warehouse. Additional parameters can be configured:

  – Use detailed data

  – Use summarized data

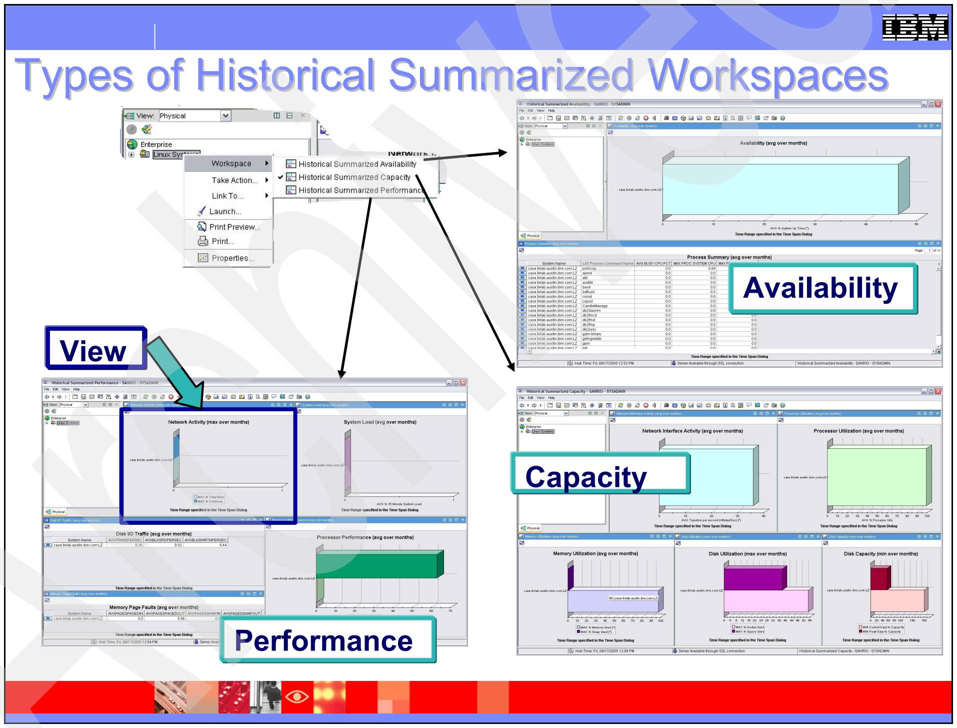
- Custom (check boxes)

  The custom parameters enable you to specify a date and time-span range, as well as use shift data and specify working days and vacation days in the report.

The agents are also ship with historical summarized workspaces. There are three primary types of historical summarized workspaces.

- Availability
- Capacity
- Performance

Figure 4-26 shows an example of the three primary types of historical summarized workspaces views available.



Types of Historical Summarized Workspaces
Workspace
Take Action...
Link To...
Launch...
Print Preview...
Print...
Properties...
Historical Summarized Availability
Historical Summarized Capacity
Historical Summarized Performance
View
Availability
Capacity
Performance

Figure 4-26 Historical summarized workspace views

The historical summarized workspace views can be used to drill down to monthly, weekly, daily, and hourly summarization periods. The top-level view is a monthly aggregrate view (that is, summarized monthly data over the selected timespan). From there, you drill down to weekly, daily, and hourly data.

5

Tivoli Distributed Monitoring V3.7 upgrade

This chapter details procedures for upgrading components of a Tivoli Distributed Monitoring 3.7 environment into an IBM Tivoli Monitoring 6.1 infrastructure. The components and structure of a Tivoli Distributed Monitoring and Tivoli Management Framework environment are outlined, then compared to the new components in an IBM Tivoli Monitoring 6.1 environment. Installation procedures for the upgrade tools are detailed, followed by an overview of the new tools and other elements that are installed in the existing Tivoli management region (TMR). Best practices are detailed for upgrading the endpoints, monitoring profiles, and profile managers in a TMR to corresponding agents, situations, and managed system lists in an IBM Tivoli Monitoring 6.1 environment. Post-upgrade considerations are then outlined for the cleanup and maintenance of the new environment.

This chapter covers:

- Comparing Tivoli Distributed Monitoring to IBM Tivoli Monitoring 6.1
- Installing the Upgrade Tool components
- Using the Upgrade Tools
- Post-upgrade considerations

5.1 Comparing Tivoli Distributed Monitoring to IBM Tivoli Monitoring 6.1

It is important from the beginning to consider the scope of the upgrade, as it implies change, both to a new monitoring environment and to a new overall infrastructure. An existing Tivoli Distributed Monitoring environment is usually just one subcomponent of a Tivoli Management Framework environment that encompasses multiple other Tivoli management applications. Thus, the current Tivoli Management Framework infrastructure layout that Tivoli Distributed Monitoring depends on may differ from the ideal corresponding IBM Tivoli Monitoring 6.1 environment.

Additionally, as noted in previous chapters, differences in scalability best practices may affect the ideal target IBM Tivoli Monitoring 6.1 infrastructure and distribution of management agents. Because of this, it is important to understand how components of Tivoli Distributed Monitoring and the Tivoli Management Framework map to IBM Tivoli Monitoring 6.1 components.

5.1.1 Infrastructure and managed resources

In the Tivoli Management Environment® (TME®), a Tivoli management region (TMR) consists of a single TMR server and typically one or more other managed nodes that may act as Gateways. Gateways act as a communications interface to the endpoints, so that all TME-based communication going to or coming from those systems is channeled through the Gateway.

Figure 5-1 shows the Tivoli Management Environment.

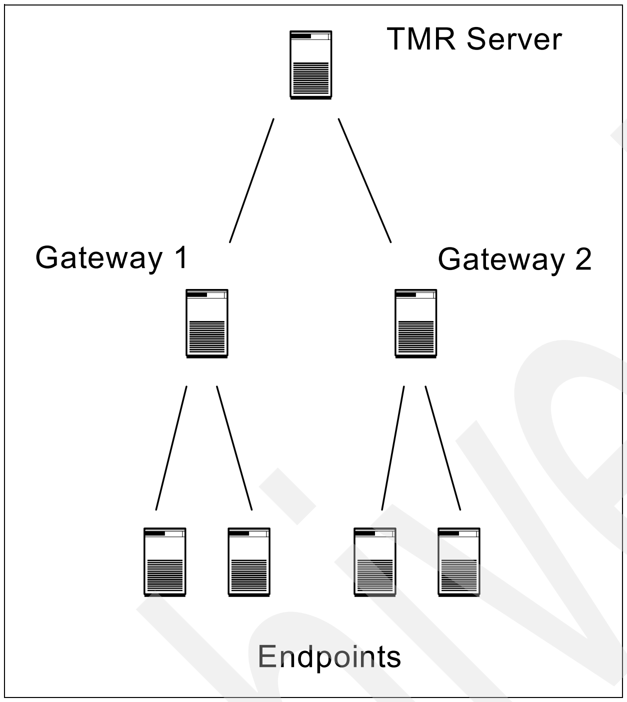


Figure 5-1 Basic TME environment

A TMR may be interconnected with other TMRs to form an interconnected TMR hierarchy, with sharing of resources in either one or both directions depending on management needs (Figure 5-2).

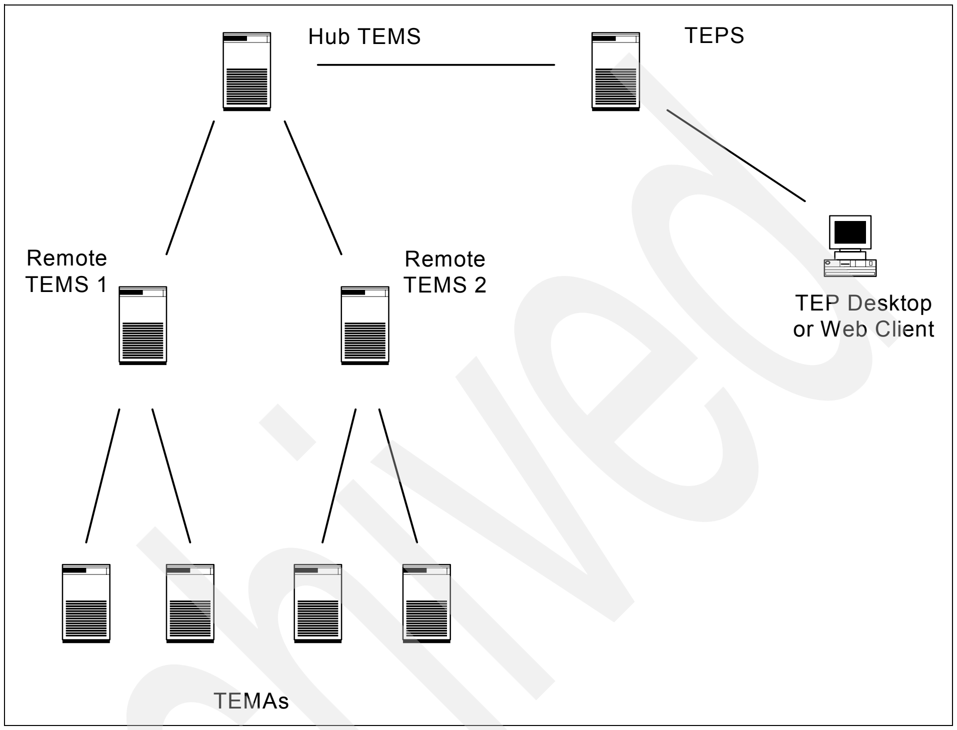


Figure 5-2 IBM Tivoli Monitoring 6.1 infrastructure, agents, and clients

In IBM Tivoli Monitoring 6.1, the Hub Tivoli Enterprise Monitoring Server (Hub TEMS) closely parallels the TMR server in terms of its role as a centralized location for managing the environment. A Remote Tivoli Enterprise Monitoring Server (Remote TEMS) handles the role of communications interface between the Hub TEMS and the monitored systems called Tivoli Enterprise Management Agents (TEMAs). Most communication going to and coming from the monitored systems is channeled through the TEMS; the exception is historical data on the monitored systems that may directly contact a component known as the Warehouse Proxy without traversing the TEMS. (This is discussed in Chapter 4, “Historical summarized data” on page 185.)

A single IBM Tivoli Monitoring 6.1 environment cannot interconnect with other environments; however, multiple Warehouse Proxies from different monitoring environments can forward historical data to a single target RDBMS to allow reporting via third-party tools of multiple monitoring environments.

5.1.2 Monitoring elements

In Distributed Monitoring, monitors are executed by the engine to acquire data related to the system. A single numeric or string value is returned by a monitor, representing a single attribute for a single instance of some resource on the system, and one or more thresholds are compared to that value to determine what responses to take. For example, a “Percent Space used” filesystem monitor would have an argument to specify one particular filesystem, and when that monitor executed, it would return a single numeric value representing the percent filesystem space used by that one filesystem. That monitor could be set up with the default thresholds of “critical,” “severe,” and “warning,” with each threshold level having one or more responses associated with it, such as sending a IBM Tivoli Enterprise Console event or running a program. Monitoring additional filesystems would require using additional monitors, one for each filesystem. Monitoring multiple similar attributes of the filesystem (for example, monitoring both percent space free and percent i-nodes free) is also only achievable using multiple monitors.

One or more monitors are grouped into profiles; profiles allow organization of monitors into groupings that can be managed as a whole. These profiles are further organized into profile managers, which have subscribers that can receive distributions of these profiles. Possible subscribers include endpoints, application proxy objects, and other profile managers. Because profile managers can subscribe to each other, a profile manager hierarchy can be set up.

To begin monitoring endpoints, a Distributed Monitoring engine is installed on the endpoint via a SentryProfile distribution; the engine is a process running separately from the endpoint (lcfd) process, but it is dependent on lcfd for communications to the TME environment via upcalls and downcalls. To monitor applications and databases, additional monitors are available in separate monitoring collections. The monitoring implementation may involve additional processes separate from the engine, but initiating monitor execution, scheduling, response handling, and providing status information is all handled by the single engine process.

In IBM Tivoli Monitoring 6.1, the entity corresponding to a monitor is called a *situation*, and these situations run in TEMAs. However, there are key differences:

- The action corresponding to a profile distribution would be the use of the Agent Deployment feature from the GUI or CLI.
- To monitor additional applications and databases, new separate TEMAs are installed for each application or database. Each TEMA uses common APIs for connectivity to the TEMS and other infrastructures, but each TEMA runs independently from the other; so scheduling, situation execution, response handling, and status information are all handled by separate TEMA

processes. So for example, to monitor a system with a UNIX OS, a group of logfiles, and a DB2 database would require running three separate agents.

- A TEMA can collect multiple types of attributes of a system, and these attributes are organized into attribute groups. Situations can examine a single attribute value if desired, but a single situation can also examine multiple attributes from the same attribute group for that particular system. For example, a single situation called Filesystem_Problems might examine the percent filesystem space free and also the percent i-nodes free, and only if both were beyond particular thresholds would a Critical event be sent to the Situation Event Console. Also, that single situation would by default evaluate all of the filesystems, not just one; the situation formula could be further limited to only monitor particular filesystems if desired.
- Embedded situations and correlated situations further allow analysis of multiple attributes from different attribute groups and from agents on different systems.
- A single situation does not include multiple threshold levels with varying response actions for each threshold; instead, separate situations can be created for separate severities to allow for varying response actions. Three severities are available: Critical, Warning, and Informational.

Additionally, the concept of profiles and profile managers is replaced by the use of managed system lists (MSLs), which group one or more agents of the same type together. These MSLs can then be assigned as distribution targets for situations. Unlike the TME-based profile manager hierarchy, MSLs use a flat one-level structure and are not nested.

Each of these key issues affects the use of the Upgrade Tools outlined later in the chapter.

5.1.3 Management and reporting

In Tivoli Management Framework and Tivoli Distributed Monitoring, CLI-based management of the monitoring infrastructure is accomplished via commands primarily done from any of the managed nodes, and via a limited subset of commands on the monitored systems themselves (Figure 5-3 on page 245). GUI-based management of the monitoring environment is accomplished through the Tivoli Desktop, and GUI-based reporting is accomplished through integration with a product external to the environment called *Tivoli Decision Support for Server Performance Prediction* (TDS/SPP in Figure 5-3).

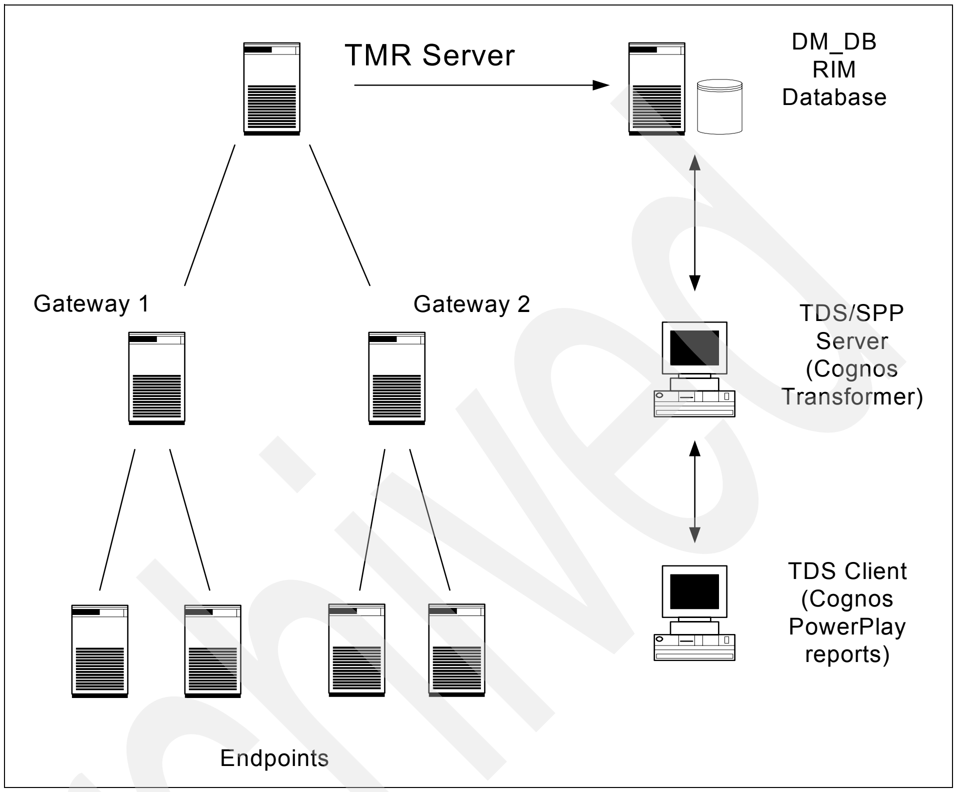

TMR Server
DM_DB RIM Database
Gateway 1
Gateway 2
TDS/SPP Server (Cognos Transformer)
TDS Client (Cognos PowerPlay reports)
Endpoints

Figure 5-3 Distributed Monitoring reporting hierarchy

Command-line management of the IBM Tivoli Monitoring 6.1 environment is done primarily from one of the TEMSs, with a limited set of commands available on the agents themselves. The Tivoli Enterprise Portal Server (TEPS) acts as the single point of interface for both GUI-based management of the monitoring environment and GUI-based reporting, via a Tivoli Enterprise Portal (TEP) Desktop Client or Web Client interface (Figure 5-4).

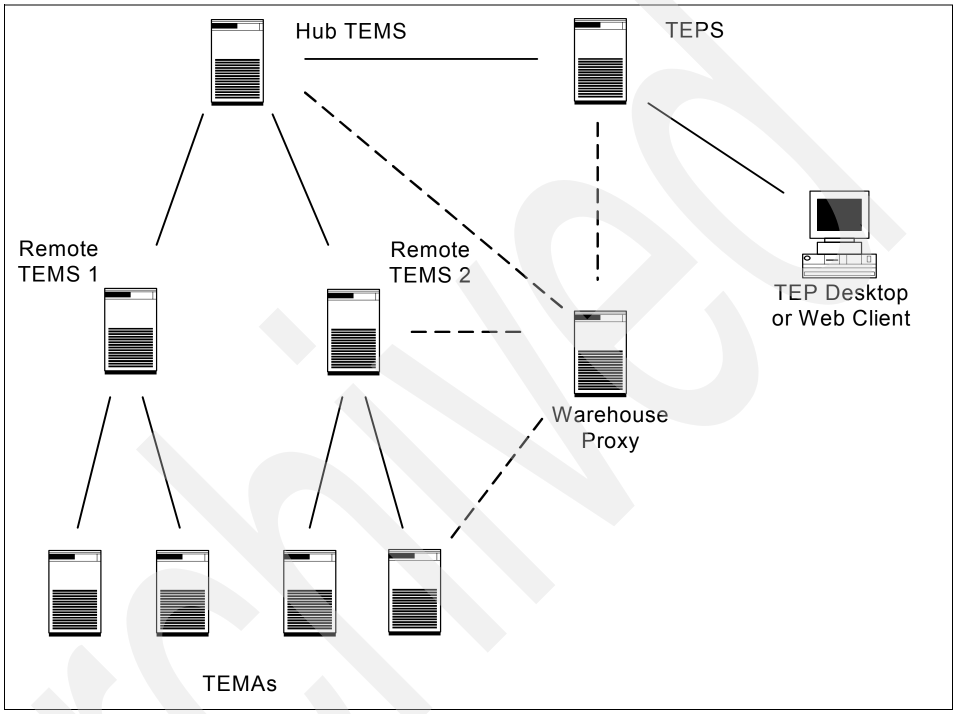


Figure 5-4 IBM Tivoli Monitoring 6.1 reporting hierarchy

5.2 Installing the Upgrade Tool components

This section details the requirements and procedures for installing the toolkit in the Tivoli Management Framework and Distributed Monitoring environment.

5.2.1 Requirements

The stated requirements for the IBM Tivoli Monitoring Upgrade Toolkit 6.1 product build that was used for this book showed:

- Framework 3.7 or higher
- Distributed Monitoring 3.7 or higher

- Java for Tivoli, Version 1.3.0 or later; or IBM Runtime Environment for Java, Version 1.4.2

The OPMT.IND product installation file showed the following dependencies:

```
opmt:depends:TMF_3.7.1
opmt:depends:Sentry2.0.2
```

This implies that Framework 3.7.1 is required. Also, it implies that versions older than 3.7 of Distributed Monitoring would be accepted by the installer, and that it is not checking for particular Distributed Monitoring Fix Pack levels. Based on testing, a recommended fix pack level would include:

- 3.7-DMN-0001
- 3.7-DMN-0003 (for Linux endpoints)
- 3.7-DMN-0020
- 3.7-DMN-FP01
- 3.7-DMN-FP04
- 3.7-DMN-0051LA, 3.7-DMN-0054LA, 3.7-DMN-0079LA (for Linux endpoints)
- 3.7-DMN-0078LA

LA patches 3.7-DMN-0078LA and 3.7-DMN-0079LA were created to handle a newly discovered defect regarding the PROFILEOID environment variable that affected the proper upgrade of custom monitors.

Also, while the Java Runtime Environment (JRE) provided by the “Java 1.3 for Tivoli” product (which is installable from the Tivoli Management Framework product installation media) is capable of running the upgrade tools, the IBM Runtime Environment for Java Version 1.4.2 is recommended due to issues related to the association of situations with navigator items to allow proper display of alerts in the TEP client.

Note: The IBM JRE 1.4.2 is available for download for AIX and Linux at:

`http://www.ibm.com/developerworks/java/jdk/index.html`

As we wrote this book, that link did not include downloads for other platforms. An alternative method for acquiring the IBM JRE 1.4.2 is to run the installer for the agent for a given platform and proceed to the point where it automatically installs the JRE, then use that JRE image. For example, running the install.sh script for UNIX Agents on a Solaris machine by default extracts the IBM JRE 1.4.2 to the /opt/IBM/ITM/JRE directory, with a sol### subdirectory underneath it (where ### indicates a version number) containing the JRE image files and directories. That directory can be archived using **`tar`** and extracted to another path if desired.

5.2.2 Installation procedure

The upgrade toolkit is shipped as three main installable products, with an additional language support product:

1. IBM Tivoli Monitoring Upgrade Toolkit 6.1: This provides the base tools, supporting files, and support for upgrading Windows endpoints.
2. IBM Tivoli Monitoring Upgrade Toolkit 6.1 - UNIX Agent Support: This provides additional support for upgrading AIX, Solaris, and HP-UX endpoints.
3. IBM Tivoli Monitoring Upgrade Toolkit 6.1 - Linux Intel Agent Support: This provides additional support for upgrading Linux endpoints. Note that although IBM Tivoli Monitoring 6.1 supports many different versions of Linux, the only officially supported version of Linux that Distributed Monitoring has in common with IBM Tivoli Monitoring 6.1 is Red Hat Enterprise Linux (RHEL) 2.1 for Intel platforms.
4. IBM Tivoli Monitoring Upgrade Toolkit 6.1 - Language Support

Note: The product installation images for several of these products are large relative to other products due to the number of agent image files that are included in the LCFNEW filepacks. For example, the Base toolkit is approximately 200 MB in size, the Linux Support around 250 MB, and the UNIX Support around 500 MB. For installation on the TMR server and Gateway connected at LAN speeds, this may not be an issue. For Gateways separated by slow links, extracting the product installation files locally and installing using those files can improve performance.

IBM Tivoli Monitoring Upgrade Toolkit 6.1 installation

This should be installed on the TMR server and on Gateways serving endpoints with Distributed Monitoring. There are two optional arguments available with this installation:

ScanValue=<value>	<value> can be 0 or 1. Determines whether to run the **`witmscantmr`** tool during the installation. The default 0 value indicates *not* to run the scan during the product installation. This is recommended to prevent difficulties in troubleshooting any installation problems that might occur, and to improve the speed of the installation.
OpmtJavaDir=<path>	<path> is the fully qualified path to the Java Runtime Environment, up to and including the bin directory. For example, /usr/jre142/java/bin. This argument is required if ScanValue=1 is used. If it is not specified, the value can be set after installation using the **`witmjavapath`** command (outlined later in the chapter).

For example, if the installation media were in the current working directory on a UNIX TMR server, the following command could be used to install onto the TMR server `bootcamp` and the managed node `moonwater`:

```
winstall -c 'pwd' -i OPMT bootcamp moonwater
```

In this case, the path to the JRE on the TMR server was not specified, and the following warning would appear during installation, during the Server Database (ALIDB) filepack installation on the TMR server:

```
The path to the Java executable was not specified. Please execute
witmjavapath after the installation completes.
```

The installation should still proceed normally, but the JRE path has to be set after installation using the **`witmjavapath`** command on the TMR server. For example:

```
witmjavapath /usr/jre142/java/bin
```

Additionally, helpful files are included in subdirectories of the BASE product installation directory:

SAMPLES	Contains XSL stylesheets that can be used to customize formatted viewing of the XML output.
SCHEMAS	Contains XML Schema definition language (XSD) files that can be used by validation parsers to validate edited XML content.
UPGINFO	Contains HTML files that include tables indicating supported and unsupported monitors for the upgrade. The upgrade tool supports the upgrade of key monitors in the Unix_Sentry, Windows 2000, and Universal monitoring collections; those that are not supported for upgrade are listed in a separate HTML file in this directory.
LOGVIEWER	Java-based utility for converting XML-based log output to HTML, to enable viewing in a browser.

UNIX Agent Support installation

This should be installed on any managed nodes that have Gateways that can service UNIX endpoints with Distributed Monitoring.

Linux Intel Agent Support installation

This should also be installed on any managed nodes that have Gateways that can service Linux endpoints with Distributed Monitoring.

Language Support installation

This should be installed on the TMR server and on any managed nodes that have the base support installed.

5.2.3 Post-install components

On the TMR server, the ALIDB filepack after-script does the following:

- Two new classes are created: ITMUpdateManager (a distinguished class for managing the upgrade of endpoints) and ITMUpgradeManagerEPA (for the downcall handlers related to the OS Agent upgrades).
- The directories shown in Example 5-1 are created and used later during the upgrade process.

Example 5-1 Directories created and used during the upgrade process

```
“$DBDIR"/AMX
$DBDIR"/AMX/trace
“$DBDIR"/AMX/logs
"$DBDIR"/AMX/data/environments
"$DBDIR"/AMX/data/mapping
"$DBDIR"/AMX/data/config
"$DBDIR"/AMX/shared/analyze/scans
“$DBDIR"/AMX/shared/analyze/profiles/upgrade
"$DBDIR"/AMX/shared/analyze/profiles/rollback
"$DBDIR"/AMX/shared/analyze/profiles/cleanup
"$DBDIR"/AMX/shared/analyze/profilemanagers/upgrade
"$DBDIR"/AMX/shared/analyze/profilemanagers/rollback
"$DBDIR"/AMX/shared/analyze/profilemanagers/cleanup
"$DBDIR"/AMX/shared/analyze/endpoints/upgrade
"$DBDIR"/AMX/shared/analyze/endpoints/rollback
"$DBDIR"/AMX/shared/analyze/endpoints/cleanup
```

- Mapping files that map Distributed Monitoring monitors to corresponding situations in IBM Tivoli Monitoring 6.1 are copied from a generic path ($BINDIR/../generic_unix/TME/ITMUpgrade/ITMUpgradeManager/MappingFiles) to the $DBDIR/AMX/data/mapping directory. These can be examined later to help determine how supported monitors will be mapped.
- An uninstall object named opmt is registered. It references an uninstall script at $BINDIR/TME/ITMUpgrade/ITMUpgradeManager/rmopmt.sh.
- Dependency Sets are created for the upgraded agents to use, and the Gateways are all synchronized with the updated dependencies using **`wgateway <GatewayOID> dbcheck`**:

ITMAgentDeploy	Associated with EPA_installOSAgent downcall
ITMAgentRemove	Associated with EPA_uninstallOSAgent downcall

ITMCustomScript	Associated with EPA_deployScript downcall
ITMEventForwarding	Associated with EPA_setEventForwarding downcall

- If the installation argument ScanValue was enabled (set to “1”), and if the JRE path was set by the OpmtJavaDir installation argument, then it will attempt to run the **`witmscantmr -c`** command. Results would be sent to the $DBDIR/AMX/shared/analyze/scans/$TMROID.xml file, where $TMROID is the region number for the TMR.

If installation failures occur, consider retrieving and analyzing the following data:

- Installation log output is stored in the wtemp directory. Run **`wtemp`** to confirm which directory this is (usually /tmp, /var/tmp, or %DBDIR%/tmp), then examine the latest directory contents sorted by timestamp using **`ls -ltr`** for a UNIX or bash environment, or **`dir /od`** for a Windows environment. Look for *.error, *.cinstall, *.output, and other files with recent timestamps. Errors during the ALIDB server database filepack install would be in an opmt_ALIDB_after.error file; the problem leading to the installation failure should be at the end of this file.
- The most common cause of failure has been using all the space on the filesystem where LCFNEW resides. On UNIX, **`df -k $BINDIR/../lcf_bundle.40`** should show what space is available on the filesystem. If all three components (Base support, UNIX support, and Linux support) are installed, it requires about 1 GB of space to install the LCFNEW binaries.

5.3 Using the Upgrade Tools

This section provides best-practices guidelines for scanning the existing TMR environment, generating a corresponding IBM Tivoli Monitoring 6.1 infrastructure; assessing the endpoints, Distributed Monitoring profiles, and profile manager hierarchies; and upgrading them to the new IBM Tivoli Monitoring 6.1 environment.

These tools are Java-based; if the JRE path was not set during installation, or if the path was set but needs to be changed, the **`witmjavapath`** command must be used on the TMR server to set the correct JRE path. To confirm the current JRE path setting, running **`witmjavapath`** without arguments shows the value. The fully qualified path up to and including the bin subdirectory should be used, where the bin directory includes the java JRE binary. For example, to specify the path to a JRE in the path /usr/java/jre142, the command would be:

```
witmjavapath /usr/java/jre142/bin
```

The command should be run on the TMR server because all upgrade commands are run from the TMR server.

5.3.1 Phase 1: Building the infrastructure

The **witmscantmr** command is used to identify the current resources in the TME infrastructure, and to build a baseline file in XML format that does an initial mapping of those resources to suggested resources in a new IBM Tivoli Monitoring 6.1 infrastructure.

Figure 5-5 includes a simple flowchart detailing high-level steps to follow for building the infrastructure.

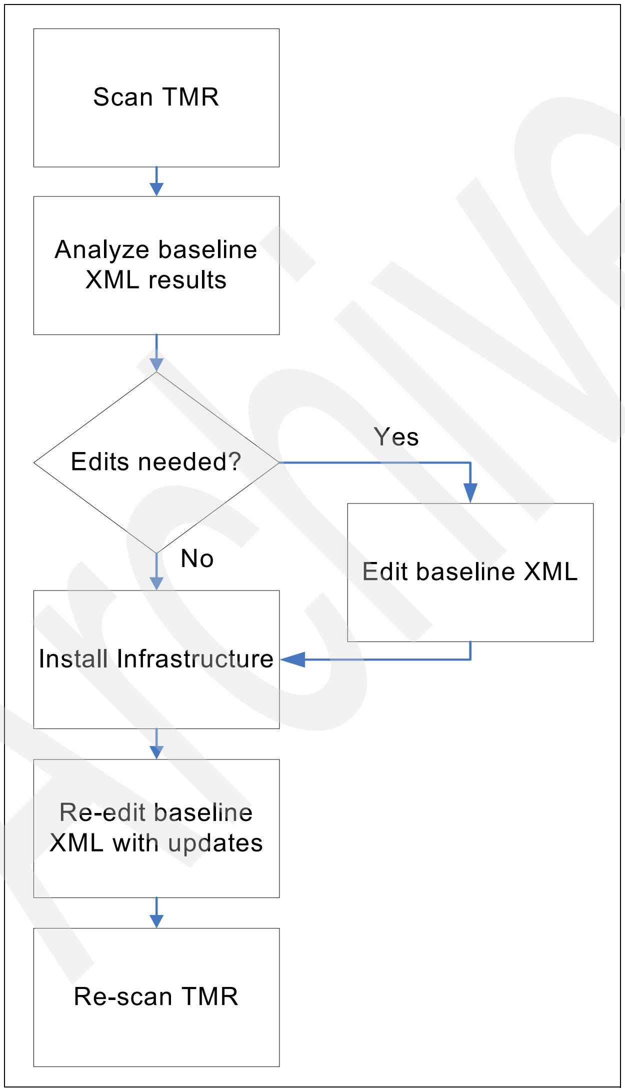


Figure 5-5 Building Infrastructure flowchart

The steps above are detailed in upcoming sections with sample output.

The environment in Figure 5-6 is used as an example for the scan, assess, analyze, and upgrade steps that follow.

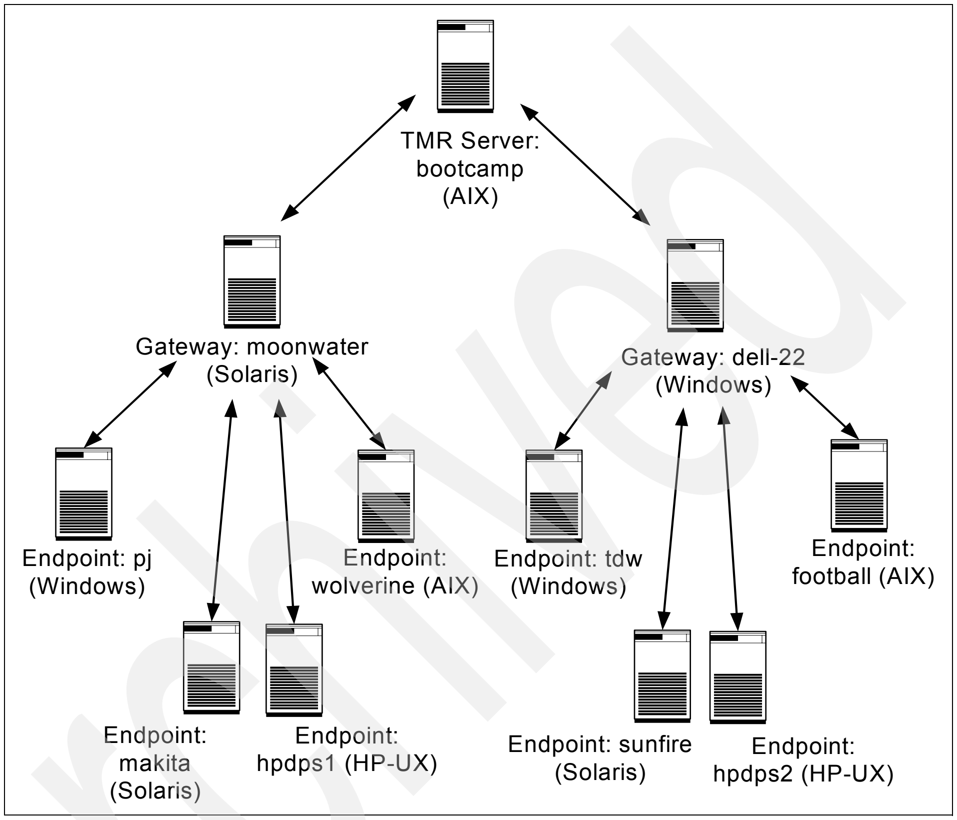

TMR Server: bootcamp (AIX)
Gateway: moonwater (Solaris)
Gateway: dell-22 (Windows)
Endpoint: pj (Windows)
Endpoint: wolverine (AIX)
Endpoint: tdw (Windows)
Endpoint: football (AIX)
Endpoint: makita (Solaris)
Endpoint: hpdps1 (HP-UX)
Endpoint: sunfire (Solaris)
Endpoint: hpdps2 (HP-UX)

Figure 5-6 Sample TME environment

Before starting to use the commands, consider enabling debug tracing by running the following lines on the TMR server, in a shell where the Tivoli environment is already sourced:

```
OID-'wlookup -o ITMUpgradeManager'
idlcall $OID _set_debug TRUE
```

Enabling debug tracing will cause additional trace logs to be generated in the $DBDIR/AMX/trace directory. Also, if debug tracing is enabled, temporary files that are normally removed are left intact.

Scan the TMR

The TMR infrastructure is scanned for the first time by running the following command from the TMR server:

```
witmscantmr -c
```

This generates the baseline XML file on the TMR server in $DBDIR/AMX/shared/analyze/scans/$TMR.xml, where $TMR is the region number. Output similar to Example 5-2 appears following a successful execution of **witmscantmr -c**:

Example 5-2 witmscantmr -c output

```
AMXUT0016I The software is processing the current request.
.....

AMXUT2600I The results of the witmscantmr command were saved to the file:
 /usr/local/Tivoli-4111/bootcamp.db/AMX/shared/analyze/scans/1438770036.xml.

AMXUT0008I The command completed successfully. Refer to

/usr/local/Tivoli-4111/bootcamp.db/AMX/logs/log_witmscantmr_20051020_15_24_01.l
og
for more information.
```

Important issues to consider in the generation of this baseline XML file:

- The **witmscantmr -c** command causes the OpmtMethods process to start. It begins by acquiring the list of IBM Tivoli Enterprise Console Event Servers; it programatically does the equivalent of **wlookup -ar EventServer**. Thus, it retrieves both local and remotely registered entries that may come from interconnected TMRs. The output is temporarily stored in a wtemp/esData_<yyymmdd_HH_MM_SS>.txt file, where <yyyymmdd_HH_MM_SS> represents a timestamp.
- It then examines the list of all known Gateways, and reduces the list to only the Gateways in the local TMR.
- It confirms whether the Gateways have Distributed Monitoring installed. If a Gateway does not have it installed, then it does not examine that particular Gateway for further details relative to the upgrade, because it assumes that there are no connected endpoints running Distributed Monitoring. However, in most environments, Distributed Monitoring is installed on all Gateways in a TMR to prevent problems with endpoints migrating to a Gateway that did not have Distributed Monitoring installed.
- For Gateways that have Distributed Monitoring installed, it then examines a list of Application Proxy objects on the Gateway. It looks for instances of objects related to the Tivoli Manager and IBM Tivoli Monitoring 5.x PAC

products such as Tivoli Manager for Oracle or IBM Tivoli Monitoring 5.x for DB2. It gets a total count of these objects that are registered on the Gateway. This is being done to provide an adjusting estimation for the total number of agents that will be expected for a corresponding Remote TEMS in the new IBM Tivoli Monitoring 6.1 environment. It is done with the anticipation that following the upgrade to the new corresponding OS Agent, other agents will be installed on the systems, and that they likely correspond to many of the Application Proxy objects that exist on the Gateway. It tries to quantitatively take into consideration the fact that, although the Upgrade tools can deploy the OS Agents corresponding to the base OS monitoring, there will likely be other non-OS Agents installed later that will extend the total count of agents.

- It then examines the list of endpoints on the Gateway and determines whether Distributed Monitoring is installed on them by looking for the Sentry_Boot_1 boot method that is registered in the endpoint manager for the endpoint when an engine is started. It programatically does the equivalent of this:

  ```
  EPOID='wlookup -or endpoint <endpointName>'
  wep boot_method list Sentry_Boot_1 $EPOID
  ```

  (<endpointName> is the name of the endpoint.) If it has this boot method, it assumes that Distributed Monitoring is installed, adds it to the list of endpoints to be upgraded, and updates the count.

- If it does not have Distributed Monitoring installed, it then determines whether IBM Tivoli Monitoring 5.x is installed by looking for the tmnt_boot boot method. If it does have ITM 5.x installed, it does not add it to the list of endpoints to be upgraded, but it still increments the count. This is being done again as an adjustment to the estimated number of agents to be assigned to the corresponding Remote TEMS. It does this to anticipate the agent growth from the use of the IBM Tivoli Monitoring V5.x Endpoint Agent that integrates IBM Tivoli Monitoring 5.x Resource Model data with the TEP.

- It then writes the names of Distributed Monitoring and ITM 5.x endpoints to a temporary file: 'wtemp'/gwData_<yyyymmdd_HH_MM_SS>.txt. To distinguish between those endpoints that actually have Distributed Monitoring, and those that only have ITM 5.x installed, the last column in each row of data in the file is used: a 1 indicates it has Distributed Monitoring; a 0 indicates it had only ITM 5.x.

- For each Gateway it follows this procedure, then updates the 'wtemp'/gwData_<yyyymmdd_HH_MM_SS>.txt file with the Gateway list. Each entry in the file consists of the Gateway name, Application Proxy count, valid endpoint count, and a flag indicating whether Distributed Monitoring is installed (1 if it is installed, 0 if not).

At this point, the OpmtMethods process spawns Java to load the scanTMR class (com.ibm.opmt.ScanTMR). This class parses the corresponding 'wtemp'/*data* files generated above to form corresponding XML. The entry `Shelling out to` `call java` appears in the trace file to indicate this key break point. Some items to note regarding the flow of this component:

- The resulting XML file will always have at least one entry for a Hub TEMS (HubServer), and at least one entry for a TEPS (PortalServerInfo). Also, it will have at least one Remote TEMS (Remote Server) assigned for each Gateway that has Distributed Monitoring installed.
- When determining how many Remote TEMS to assign, there are two parameters read from a scanTMR.properties file embedded within the scantmr.jar file:
  - MaxHeartbeatsPerRemote: This is the maximum number of agents that will be assigned to a particular Remote TEMS. This value is set to 500, correlating to the suggested value mentioned in the Product Architecture chapter earlier.
  - MaxRemotesPerHub: This is the maximum number of Remote TEMS that will be created in a particular Hub, and is set to 15.
- The formula for the number of Remote TEMS required for a particular Gateway is approximately the following:

  ```
  [AppProxyCount + ( 2 * endpointCount ) / MaxHeartbeatsPerRemote]
  ```

  where `AppProxyCount` is the count of Application Proxy objects mentioned earlier in the gwData*.txt file, `endpointCount` is the count of endpoints with either Distributed Monitoring or IBM Tivoli Monitoring 5.x installed on them, and `MaxHeartbeatsPerRemote` is 500. Again, the AppProxyCount is added as an adjustment to account for future agents that would likely be installed following the upgrade. Additionally, the endpointCount is multiplied by two to account for use of additional agents beyond the base OS Agent and beyond the application agents mentioned earlier such as the Universal Agent. This provides a conservative value to estimate the number of Remote TEMS to handle the load of both the upgraded OS Agents *and* the anticipated non-OS Agents that would be installed afterward.
- The total endpointCount is divided by the number of Remote TEMS to get the number of agents per remote TEMS for the Gateway. So for the baseline, the starting assumption is for a relatively even distribution of agents across the Remote TEMS.
- If the total number of Remote TEMS exceeds 15, a new Hub TEMS configuration is added to the baseline XML to account for the growth.

Analyze baseline XML results

The resulting baseline XML provides a template that should be analyzed to confirm that the infrastructure meets anticipated needs. In the example above, a single HubServer entry is made since the total number of anticipated agents was small. Example 5-3 on page 257 shows an excerpt of the RemoteServer entries.

Example 5-3   excerpt of the “RemoteServer” entries

```
<RemoteServer aggregateStatus="INCOMPLETE" hostname="server0.remote.com:1918"
protocol="IP.PIPE" source="Gateway:bootcamp4111-GW" status="NOT_DEPLOYED"
target="Remote:REMOTE_Server0" unix_OSAgent_installDir=""
windows_OSAgent_installDir=""/>
      <RemoteServer aggregateStatus="INCOMPLETE"
hostname="server1.remote.com:1918" protocol="IP.PIPE"
source="Gateway:dell-224111-GW" status="NOT_DEPLOYED"
target="Remote:REMOTE_Server1" unix_OSAgent_installDir=""
windows_OSAgent_installDir=""/>
      <RemoteServer aggregateStatus="INCOMPLETE"
hostname="server2.remote.com:1918" protocol="IP.PIPE"
source="Gateway:moonwater4111-GW" status="NOT_DEPLOYED"
target="Remote:REMOTE_Server2" unix_OSAgent_installDir=""
windows_OSAgent_installDir=""/>
```

Thus, it automatically added a RemoteServer entry for each Gateway.

Example 5-4 shows an excerpt of the baseline XML output.

Example 5-4   Sample baseline XML output

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Copyright IBM Corporation 2005-->
<ITM6.1Infrastructure aggregateStatus="INCOMPLETE"
xmlns="http://www.ibm.com/Tivoli/ITM/scantmr"
xmlns:itmst="http://www.ibm.com/Tivoli/ITM/MigrateStatus"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.ibm.com/Tivoli/ITM/scantmr tmr.xsd">
   <HubServer aggregateStatus="INCOMPLETE" hostname="server0.hub.com:1918"
hub_installDir="" protocol="IP.PIPE" source="ServerManagedNode:bootcamp"
status="NOT_DEPLOYED" target="Hub:HUB_Server0" tec_forwarding_endpoint=""
unix_OSAgent_installDir="/opt/IBM/ITM/"
windows_OSAgent_installDir="C:\IBM\ITM\">
      <SOAPConnectionInfo SOAPencoding="" hostname="server0.hub.com:1920"
password="" status="NOT_DEPLOYED" user="SYSADMIN"/>
      <PortalServerInfo hostname="portalserver0.portal.com:1920" password=""
status="NOT_DEPLOYED" user="SYSADMIN"/>
      <EventServerList eventServerLabel="EventServer#scary-region"
eventServerTarget=""/>
      <EventServerList eventServerLabel="spoketec" eventServerTarget=""/>
      <EventServerList eventServerLabel="EventServer#bootcamp4111-region"
eventServerTarget=""/>
      <RemoteServer aggregateStatus="INCOMPLETE"
hostname="server4.remote.com:1918” protocol="IP.PIPE"
source="Gateway:bootcamp4111-GW" status="NOT_DEPLOYED"
target="Remote:REMOTE_Server4" unix_OSAgent_installDir=""
windows_OSAgent_installDir="">
```

```
        <OSAgent hostname="tdw.austin.ibm.com" interp="w32-ix86" locale="US"
osAgent_installDir="" source="endpoint:tdw" status="NOT_DEPLOYED"
target="ManagedSystem:tdw"/>
        <OSAgent hostname="bootcamp.tivlab.austin.ibm.com" interp="aix4-r1"
locale="US" osAgent_installDir="" source="endpoint:football"
status="NOT_DEPLOYED" target="ManagedSystem:bootcamp"/>
        <OSAgent hostname="hpdps1.tivlab.austin.ibm.com" interp="hpux10"
locale="US" osAgent_installDir="" source="endpoint:hpdps1"
status="NOT_DEPLOYED" target="ManagedSystem:hpdps1"/>
      </RemoteServer>
...(more RemoteServer and OSAgent entries appeared)...
```

Edit baseline XML if needed

Example 5-4 on page 257 shows:

- A single Hub TEMS is listed with default settings
- Associated SOAP Configuration and TEPS settings listed with defaults
- An Event Server List with the list of IBM Tivoli Enterprise Console Event Servers found from the Name Registry
- Only one RemoteServer entry in the example, but in the full file, multiple RemoteServer entries

In each RemoteServer entry, OSAgent entries appeared, corresponding to upgradeable endpoints that had Distributed Monitoring installed.

Decisions to make at this stage regarding removing or adding entries include:

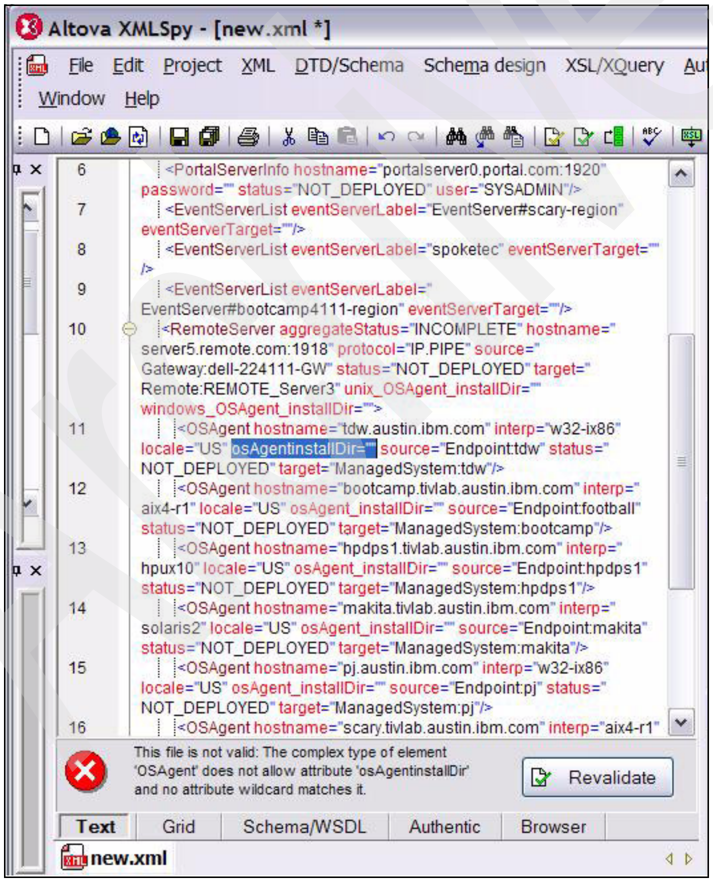
- Are there too many or too few Remote Servers for the desired new IBM Tivoli Monitoring 6.1 infrastructure? The current baseline XML is only a suggested guideline. The number of Remote TEMS can be adjusted by editing the RemoteServer entries to remove them, or to add additional ones following the same syntax used for the current entries. If they are being removed, be sure to first move the desired OSAgent entries within that RemoteServer tree, to avoid missing the upgrade of those OSAgent entries. If a Remote TEMS entry is being added, be sure to follow the same syntax used for the other entries already in the baseline file.

  In the example above, it was noted that at least one Remote TEMS is being created for each Gateway. In the new environment, this many Remote TEMS may not actually be necessary. So, the total number of Remote TEMS will likely be reduced if the count of endpoints being upgraded is small enough.
- Are all listed OSAgents really being considered for upgrade from Distributed Monitoring? This is a good time to review the listed OSAgent entries and consider whether an upgrade of the Distributed Monitoring configuration is actually necessary or desired. If it is clear that certain OSAgent entries

correspond to endpoints that do not require having their Distributing Monitoring configuration upgraded (or should not be updated for other reasons), those entries can be removed.

- Is the distribution of OSAgents across RemoteServer entries suitable? OSAgent entries can be moved among the RemoteServer entries to redistribute the anticipated load.

When editing XML, using an XML validation parser is recommended to ensure validity of the changes. In the BASE directory of the Base product support, there is a SCHEMAS subdirectory. This contains the XML Schema definition language (XSD) documents associated with the baseline XML file, and later XML files created by the Upgrade tools that can be used with a validation parser to confirm XML validity. Validation parsers use the XSD information to catch errors in spelling, attribute names, values, proper order, and general misformatting. Figure 5-7 shows an example from an XML validation parser tool.


Altova XMLSpy - [new.xml *]
File Edit Project XML DTD/Schema Schema design XSL/XQuery Au
Window Help
Revalidate
Text Grid Schema/WSDL Authentic Browser
new.xml

```
<PortalServerInfo hostname="portalserver0.portal.com:1920"
password="" status="NOT_DEPLOYED" user="SYSADMIN"/>
<EventServerList eventServerLabel="EventServer#scary-region"
eventServerTarget=""/>
<EventServerList eventServerLabel="spoketec" eventServerTarget=""
/>
<EventServerList eventServerLabel="
EventServer#bootcamp4111-region" eventServerTarget=""/>
<RemoteServer aggregateStatus="INCOMPLETE" hostname="
server5.remote.com:1918" protocol="IP.PIPE" source="
Gateway:dell-224111-GW" status="NOT_DEPLOYED" target="
Remote:REMOTE_Server3" unix_OSAgent_installDir=""
windows_OSAgent_installDir="">
<OSAgent hostname="tdw.austin.ibm.com" interp="w32-ix86"
locale="US" osAgentinstallDir="" source="Endpoint:tdw" status="
NOT_DEPLOYED" target="ManagedSystem:tdw"/>
<OSAgent hostname="bootcamp.tivlab.austin.ibm.com" interp="
aix4-r1" locale="US" osAgent_installDir="" source="Endpoint:football"
status="NOT_DEPLOYED" target="ManagedSystem:bootcamp"/>
<OSAgent hostname="hpdps1.tivlab.austin.ibm.com" interp="
hpux10" locale="US" osAgent_installDir="" source="Endpoint:hpdps1"
status="NOT_DEPLOYED" target="ManagedSystem:hpdps1"/>
<OSAgent hostname="makita.tivlab.austin.ibm.com" interp="
solaris2" locale="US" osAgent_installDir="" source="Endpoint:makita"
status="NOT_DEPLOYED" target="ManagedSystem:makita"/>
<OSAgent hostname="pj.austin.ibm.com" interp="w32-ix86"
locale="US" osAgent_installDir="" source="Endpoint:pj" status="
NOT_DEPLOYED" target="ManagedSystem:pj"/>
<OSAgent hostname="scary.tivlab.austin.ibm.com" interp="aix4-r1"
```

This file is not valid: The complex type of element 'OSAgent' does not allow attribute 'osAgentinstallDir' and no attribute wildcard matches it.

Figure 5-7 XML Validation Parser example showing an error

For the baseline XML file used here, the content is being reduced to have just one Remote TEMS, with all eight OSAgent entries assigned to it. The simplified XML file was validated in an XML validation parser and now appears as in Example 5-5.

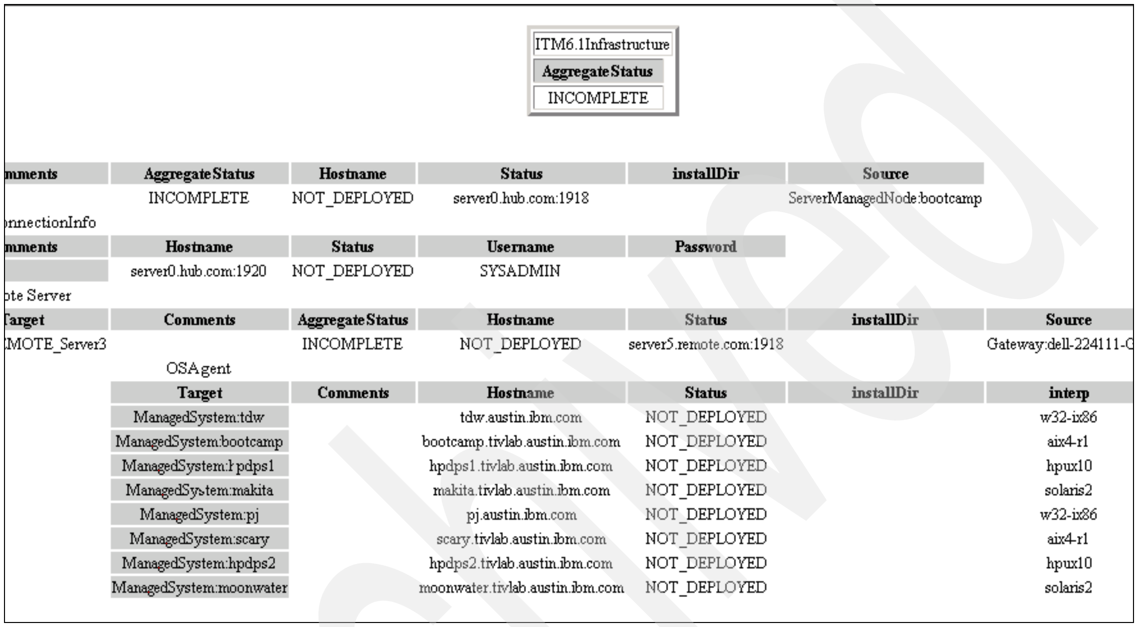
Example 5-5 New baseline XML file

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Copyright IBM Corporation 2005-->
<ITM6.1Infrastructure aggregateStatus="INCOMPLETE"
xmlns="http://www.ibm.com/Tivoli/ITM/scantmr"
xmlns:itmst="http://www.ibm.com/Tivoli/ITM/MigrateStatus" xmlns:xsi="
http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.ibm.com/Tivoli/ITM/scantmr tmr.xsd">
   <HubServer aggregateStatus="INCOMPLETE" hostname="server0.hub.com:1918"
hub_installDir="" protocol="IP.PIPE" source="ServerManagedNode:bootcamp"
status="NOT_DEPLOYED" target="Hub:HUB_Server0" tec_forwarding_endpoint=""
unix_OSAgent_installDir="/opt/IBM/ITM/"
windows_OSAgent_installDir="C:\IBM\ITM\">
      <SOAPConnectionInfo SOAPencoding="" hostname="server0.hub.com:1920"
password="" status="NOT_DEPLOYED" user="SYSADMIN"/>
      <PortalServerInfo hostname="portalserver0.portal.com:1920" password=""
status="NOT_DEPLOYED" user="SYSADMIN"/>
      <EventServerList eventServerLabel="EventServer#scary-region"
eventServerTarget=""/>
      <EventServerList eventServerLabel="spoketec" eventServerTarget=""/>
      <EventServerList eventServerLabel="EventServer#bootcamp4111-region"
eventServerTarget=""/>
     <RemoteServer aggregateStatus="INCOMPLETE"
hostname="server5.remote.com:1918"
protocol="IP.PIPE" source="Gateway:dell-224111-GW" status="NOT_DEPLOYED"
target="Remote:REMOTE_Server3" unix_OSAgent_installDir=""
windows_OSAgent_installDir="">
         <OSAgent hostname="tdw.austin.ibm.com" interp="w32-ix86" locale="US"
osAgent_installDir="" source="endpoint:tdw" status="NOT_DEPLOYED"
target="ManagedSystem:tdw"/>
         <OSAgent hostname="bootcamp.tivlab.austin.ibm.com" interp="aix4-r1"
locale="US" osAgent_installDir="" source="endpoint:football"
status="NOT_DEPLOYED" target="ManagedSystem:bootcamp"/>
......(more OSAgent entries skipped)......
</RemoteServer>
   </HubServer>
</ITM6.1Infrastructure>
```

For formatting purposes, the XML file can be converted to an HTML format using XSL stylesheets. The Base support installation media includes *.xsl files in the BASE/SAMPLES subdirectory; the tmr.xsl file is a sample stylesheet for the baesline XML file. A simple example is to place the new baseline XML file and the SAMPLES\tmr.xsl file in the same directory, then add the following line after the `<?xml version . . .` line in the new baseline XML file:

```
<?xml-stylesheet type="text/xsl" href="tmr.xsl"?>
```

Loading this into a browser that is capable of handling the formatted XML (such as Microsoft Internet Explorer) displays it as shown in Figure 5-8.



ITM6.1Infrastructure
AggregateStatus
INCOMPLETE

mments	AggregateStatus	Hostname	Status	installDir	Source
	INCOMPLETE	NOT_DEPLOYED	server0.hub.com:1918		ServerManagedNode:bootcamp

onnectionInfo

mments	Hostname	Status	Username	Password
	server0.hub.com:1920	NOT_DEPLOYED	SYSADMIN	

ote Server

Target	Comments	AggregateStatus	Hostname	Status	installDir	Source
MOTE_Server3		INCOMPLETE	NOT_DEPLOYED	server5.remote.com:1918		Gateway:dell-224111-C

OSAgent

Target	Comments	Hostname	Status	installDir	interp
ManagedSystem:tdw		tdw.austin.ibm.com	NOT_DEPLOYED		w32-ix86
ManagedSystem:bootcamp		bootcamp.tivlab.austin.ibm.com	NOT_DEPLOYED		aix4-r1
ManagedSystem:hpdps1		hpdps1.tivlab.austin.ibm.com	NOT_DEPLOYED		hpux10
ManagedSystem:makita		makita.tivlab.austin.ibm.com	NOT_DEPLOYED		solaris2
ManagedSystem:pj		pj.austin.ibm.com	NOT_DEPLOYED		w32-ix86
ManagedSystem:scary		scary.tivlab.austin.ibm.com	NOT_DEPLOYED		aix4-r1
ManagedSystem:hpdps2		hpdps2.tivlab.austin.ibm.com	NOT_DEPLOYED		hpux10
ManagedSystem:moonwater		moonwater.tivlab.austin.ibm.com	NOT_DEPLOYED		solaris2

Figure 5-8 New baseline XML formatted using XSL

Either the XML or the formatted output can be saved, printed, and used as a reference in the next step where the infrastructure is actually installed. As we wrote this book, the XSL stylesheets were being updated to handle formatting problems (as seen in the example above), and additional functioning XSL stylesheets were being planned for the other upgrade tool XML output that follows in later sections.

Install the infrastructure

Based on the resulting new baseline XML file, the Hub TEMS, TEPS, and Remote TEMS can be installed. As the current baseline XML file is only a template, the initial information present for most of the attributes is simply placeholder information that should be replaced following installation of the infrastructure.

In the sample environment used for this chapter, a Linux machine named x226 is used for the Hub TEMS. Because of the limited size, it also serves as the TEPS. A system called dell-22 acts as a Remote TEMS for the upgraded agents, as shown in Figure 5-9.

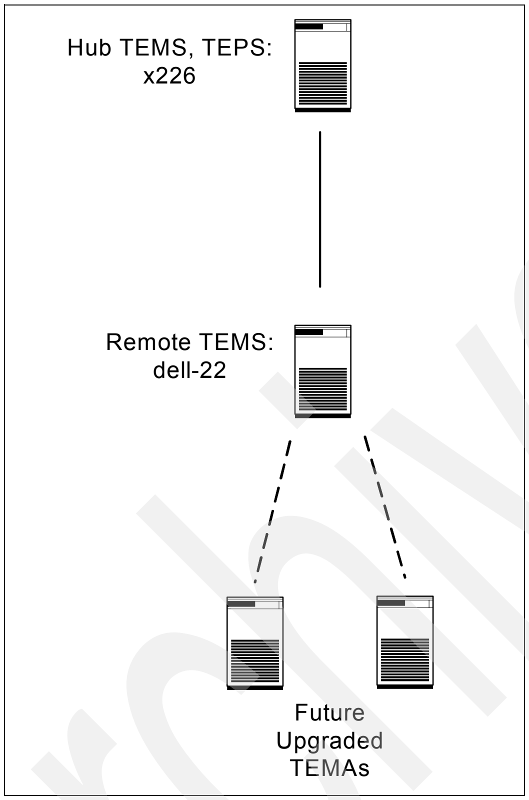


Figure 5-9 New installed infrastructure

For the Hub TEMS machine, the environment is installed using the default port (1918), protocol (IP.PIPE), naming conventions (HUB_x226), and user/password (SYSADMIN/no password), to make initial baseline XML editing easier in the next step. The Hub TEMS has the TEC Synchronization feature configured via the "Manage Tivoli Services" GUI tool, and the IBM Tivoli Enterprise Console Server has the TEC event synchronization component (Situation Update Forwarder) installed and configured per instructions in the *IBM Tivoli Monitoring Installation and Setup Guide*, GC32-9407.

Additionally, on the Hub TEMS, if TEC event synchronization is desired to be used, then a TME endpoint must first be installed on the Hub TEMS. This is to allow the other Upgrade tools used later to remotely access the Hub TEMS machine and modify a tecserver.txt file that the Hub uses when forwarding events to the appropriate IBM Tivoli Enterprise Console Servers. See Chapter 10, “Integration with IBM Tivoli Enterprise Console” on page 583 for additional information regarding TEC event synchronization.

Re-edit baseline XML with updates

For the Hub TEMS, SOAP interface, TEPS, Remote TEMS, and OS Agents, corresponding entries in the baseline XML must be updated with proper values as they are each installed. Throughout the process, the **`witmscantmr`** command can be rerun with special options to review the current environments and confirm whether they are properly deployed.

Throughout the XML, each element has a number of attributes to be reviewed, but the remaining entries should not be updated. To ease understanding of the contents, the tables below summarize only the entries that should be reviewed, updated, or both, grouped by component and element. The attributes that automatically change as the infrastructure is installed will be explained later.

First, the HubServer and RemoteServer element attributes that require review are listed in Table 5-1.

Table 5-1 HubServer and RemoteServer element attributes to review

XML attribute	Definition	Example
hostname	Fully qualified name and port separated by a colon	**x226.tivlab.austin.ibm.com:1918**
hub_InstallDir	(HubServer only) $CANDLEHOME; the base installation directory on the TEMS	**/opt/IBM/ITM**
protocol	Primary configured communications protocol	**IP.PIPE**
target	Format: hub:<HUB_name> or remote:<REMOTE_name>	Hub Example: **Hub:HUB_x226** Remote Example: **Remote:REMOTE_dell-22**
tec_forwarding_endpoint	(HubServer only) Name of endpoint installed on Hub TEMS used by later Upgrade tools to update event settings with TEC event synchronization component	For this test environment: x226

XML attribute	Definition	Example
unix_OSAgent_installDir	Default UNIX OS Agent installation path to use, but only if the OSAgent does not have its own installDir specified.	/opt/IBM/ITM
windows_OSAgent_installDir	Default Windows installation path to use, but only if the OSAgent does not have its own installDir specified.	C:\IBM\ITM

Next are the SOAPConnectionInfo element settings, which are the settings used by the SOAP server on the Hub TEMS, as shown in Table 5-2.

Table 5-2 SOAPConnectionInfo element attributes to review

XML Attribute	Definition	Example
hostname	Hub TEMS <hostname>:<port>, where <port> is 1920 for IP.PIPE and UDP protocols, and 3660 for IP.SPIPE	x226.tivlab.austin.ibm.com:1920
user	The username used to login to the Hub TEMS	SYSADMIN
password	The password associated with the above username	<no password used by default>

Next are the TEPS element settings to review and update if needed (Table 5-3 on page 257).

Table 5-3 PortalServerInfo element attributes to review

XML Attribute	Definition	Example
hostname	<hostname>:<port> for TEPS	x226.tivlab.austin.ibm.com:1920
user	The username used to login to the TEPS	SYSADMIN
password	The password associated with the above username	<no password used in this example>

Rescan the TMR

Following the editing of the XML to update the correct settings for the infrastructure components, **witmscantmr** can be rerun, and the resulting XML reexamined to determine whether the updates are valid so far. So for example, assume that the original XML example above has been edited with the new Hub TEMS, SOAP, TEPS, and Remote TEMS attributes as shown in Example 5-6. (The settings that needed to be reviewed or updated appear in bold text and correspond to the table entries listed earlier.)

Example 5-6 New baseline XML with Hub TEMS and SOAP attribute updates

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Copyright IBM Corporation 2005-->
<ITM6.1Infrastructure aggregateStatus="INCOMPLETE"
xmlns="http://www.ibm.com/Tivoli/ITM/scantmr"
xmlns:itmst="http://www.ibm.com/Tivoli/ITM/MigrateStatus"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.ibm.com/Tivoli/ITM/scantmr tmr.xsd">
<HubServer aggregateStatus="INCOMPLETE"
hostname="x226.tivlab.austin.ibm.com:1918" hub_installDir="/opt/IBM/ITM"
protocol="IP.PIPE" source="ServerManagedNode:bootcamp" status="NOT_DEPLOYED"
target="Hub:HUB_x226" tec_forwarding_endpoint="x226"
unix_OSAgent_installDir="/opt/IBM/ITM/"
windows_OSAgent_installDir="C:\IBM\ITM\">
<SOAPConnectionInfo SOAPencoding="" hostname="x226.tivlab.austin.ibm.com:1920"
password="" status="NOT_DEPLOYED" user="SYSADMIN"/>
<PortalServerInfo hostname="x226.tivlab.austin.ibm.com:1920" password=""
status="NOT_DEPLOYED" user="SYSADMIN"/>
<EventServerList eventServerLabel="EventServer#scary-region"
eventServerTarget=""/> <EventServerList eventServerLabel="spoketec"
eventServerTarget=”"/>
      <EventServerList eventServerLabel="EventServer#bootcamp4111-region"
eventServerTarget=""/>
     <RemoteServer aggregateStatus="INCOMPLETE"
hostname="dell-22.tivlab.austin.ibm.com:1918" protocol="IP.PIPE"
source="Gateway:dell-224111-GW" status="NOT_DEPLOYED"
target="Remote:REMOTE_dell-22" unix_OSAgent_installDir="/opt/IBM/ITM"
windows_OSAgent_installDir="C:\IBM\ITM">
```

Using this new XML file, another scan can be done to verify whether the specified updated settings are working as planned. For example, if this new XML file is named infrastructure.xml and is in the current working directory, the command to reverify the edited XML would be:

```
witmscantmr -v -f infrastructure.xml
```

This is the first time the tool is actually making a connection to the IBM Tivoli Monitoring 6.1 environment. It does so to verify the Hub TEMS, Remote TEMS, and TEPS login information and settings, and also to verify the SOAP connection. It is important to verify that this portion is deployed correctly first, because this tool and the later Upgrade tools will use the SOAP connection to access the new environment, assess the components, and verify what has actually been deployed.

Example 5-7 shows sample output that appears after the command is run.

Example 5-7 Sample output from witmscantmr -v -f <filename>

```
AMXUT0016I The software is processing the current request.
...........

AMXUT2600I The results of the witmscantmr command were saved to the file:

/usr/local/Tivoli-4111/bootcamp.db/AMX/shared/analyze/scans/infrastructure_2005
1022_16_06_19.xml.

AMXUT0010W The command completed with warnings. Refer to the
/usr/local/Tivoli-4111/bootcamp.db/AMX/logs/log_witmscantmr_20051022_16_06_19.l
og
log for more information.
```

In Example 5-8, the new infrastructure_20051022_16_06_19.xml file that is mentioned above is updated with the following content.

Example 5-8 Updated XML results from witmscantmr -v -f <XML>

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Copyright IBM Corporation 2005-->
<ITM6.1Infrastructure aggregateStatus="PARTIAL"
xmlns="http://www.ibm.com/Tivoli/ITM/scantmr"
xmlns:itmst="http://www.ibm.com/Tivoli/ITM/MigrateStatus"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.ibm.com/Tivoli/ITM/scantmr tmr.xsd">
   <HubServer aggregateStatus="PARTIAL"
hostname="x226.tivlab.austin.ibm.com:1918" hub_installDir="/opt/IBM/ITM"
protocol="IP.PIPE" source="ServerManagedNode:bootcamp" status="DEPLOYED"
target="Hub:HUB_x226" tec_forwarding_endpoint="x226"
unix_OSAgent_installDir="/opt/IBM/ITM" windows_OSAgent_installDir="C:\IBM\ITM">
      <SOAPConnectionInfo SOAPencoding=""
hostname="x226.tivlab.austin.ibm.com:1920" password="" status="DEPLOYED"
user="SYSADMIN"/>
      <PortalServerInfo hostname="x226.tivlab.austin.ibm.com:1920" password=""
status="DEPLOYED" user="SYSADMIN"/>
<EventServerList eventServerLabel="EventServer#scary-region"
eventServerTarget=""/>
      <EventServerList eventServerLabel="spoketec" eventServerTarget=""/>
      <EventServerList eventServerLabel="EventServer#bootcamp4111-region"
eventServerTarget=""/>
      <RemoteServer aggregateStatus="PARTIAL"
hostname="dell-22.tivlab.austin.ibm.com:1918" protocol="IP.PIPE"
source="Gateway:dell-224111-GW" status="DEPLOYED"
target="Remote:REMOTE_dell-22" unix_OSAgent_installDir="/opt/IBM/ITM"
windows_OSAgent_installDir="C:\IBM\ITM">
```

The attributes that are updated are printed in bold text above; these include the status and aggregateStatus attributes. The `status` attribute shows the status of the individual element, while `aggregateStatus` shows the overall status of the element and all nested elements below it. At this stage, proper status and aggregateStatus output would be considered the following:

ITM6.1Infrastructure	aggregateStatus=PARTIAL
HubServer	aggregateStatus=PARTIAL, status=DEPLOYED
SOAPConnectionInfo	status=DEPLOYED
PortalServerInfo	status=DEPLOYED
RemoteServer	aggregateStatus=PARTIAL, status=DEPLOYED

If the aggregateStatus still shows INCOMPLETE or the status still shows NOT_DEPLOYED for the items above, there will be a new `comment` attribute added to the affected element. For example, if a Remote TEMS was not confirmed in the resulting XML, a comment attribute similar to Example 5-9 would appear.

Example 5-9 Example comment attribute for RemoteServer element

```
...
<RemoteServer aggregateStatus="INCOMPLETE" comment="AMXUT2536W The remote
server was not found.&#xa;&#xa;Explanation: &#xa;The remote server is not
running or the specified host name is incorrect.&#xa;&#xa;Operator Response:
&#xa;Verify that the Remote Server host name is correct and that the Remote
Server is running." hostname="incorrect.notreal.co:1918" protocol="IP.PIPE"
source="Gateway:aus54-GW" status="NOT_DEPLOYED" target="Remote:REMOTE_wrong"
unix_OSAgent_installDir="" windows_OSAgent_installDir="">
...
```

If it is believed that the corresponding infrastructure is running and available, but errors still appear in the XML or the status and aggregateStatus attributes do not update with appropriate values, verify the following:

- Reverify that the infrastructure components are actually running; use Manage Tivoli Monitoring Services for Windows and UNIX, or **`cinfo`** for UNIX.
- Verify the updated attributes to confirm that the correct values were used for the host names, ports, protocols, user, and password.

- Confirm that the host names used are resolvable from the TMR server.
- Confirm valid port and protocol settings using the Manage Tivoli Monitoring Services GUI utility for each infrastructure component.
- Verify that user logon works to the TEPS using a TEP desktop or browser client from the TMR server.
- Note also that network connectivity between the TMR server and the Hub TEMS / SOAP server must be allowed in order for the status to be properly retrieved and updated.

The final step in this phase is to update any EventServerList elements for the eventServerTarget attributes. For example, the sample environment above had the three entries shown in Example 5-10.

Example 5-10 EventServerList element entries

```
<EventServerList eventServerLabel="EventServer#scary-region"
eventServerTarget=""/>
<EventServerList eventServerLabel="spoketec" eventServerTarget=""/>
<EventServerList eventServerLabel="EventServer#bootcamp4111-region"
eventServerTarget=""/>
```

The eventServerTarget attribute must be specified manually for each IBM Tivoli Enterprise Console Server. The format is:

```
<hostname>:<port>
```

<hostname> is the name of the IBM Tivoli Enterprise Console Server machine and <port> is the listening port. The hostname can be a short name, fully qualified hostname, or IP address; whatever is used must be resolvable and reachable from the Hub TEMS machine. The port is usually 0 for UNIX systems because the portmapper is used to identify it; the port is 5529 by default on Windows but it is configurable. The port used can be confirmed on the IBM Tivoli Enterprise Console Server by viewing the tec_recv_agent_port setting in the $BINDIR/TME/TEC/.tec_config file.

For the example above, the eventServertTarget attributes are updated as shown in Example 5-11.

Example 5-11 eventServertTarget attributes

```
<EventServerList eventServerLabel="EventServer#scary-region"
eventServerTarget="scary.tivlab.austin.ibm.com:0"/>
<EventServerList eventServerLabel="spoketec"
eventServerTarget="spoke2.tivlab.raleigh.ibm.com:5529"/>
<EventServerList eventServerLabel="EventServer#bootcamp4111-region"
eventServerTarget="bootcamp.tivlab.austin.ibm.com:0"/>
```

This leaves us with a baseline XML file with all of the infrastructure elements and EventServerList elements updated (Example 5-12).

Example 5-12 Final baseline XML file example after scans and edits

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Copyright IBM Corporation 2005-->
<ITM6.1Infrastructure aggregateStatus="PARTIAL"
xmlns="http://www.ibm.com/Tivoli/ITM/scantmr"
xmlns:itmst="http://www.ibm.com/Tivoli/ITM/MigrateStatus"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.ibm.com/Tivoli/ITM/scantmr tmr.xsd">
   <HubServer aggregateStatus="PARTIAL"
hostname="x226.tivlab.austin.ibm.com:1918" hub_installDir="/opt/IBM/ITM"
protocol="IP.PIPE" source="ServerManagedNode:bootcamp" status="DEPLOYED"
target="Hub:HUB_x226" tec_forwarding_endpoint="x226"
unix_OSAgent_installDir="/opt/IBM/ITM" windows_OSAgent_installDir="C:\IBM\ITM">
      <SOAPConnectionInfo SOAPencoding=""
hostname="x226.tivlab.austin.ibm.com:1920" password="" status="DEPLOYED"
user="SYSADMIN"/>
      <PortalServerInfo hostname="x226.tivlab.austin.ibm.com:1920" password=""
status="DEPLOYED" user="SYSADMIN"/>
      <EventServerList eventServerLabel="EventServer#scary-region"
eventServerTarget="scary.tivlab.austin.ibm.com:0"/>
      <EventServerList eventServerLabel="spoketec"
eventServerTarget="spoke2.tivlab.raleigh.ibm.com:5529"/>
      <EventServerList eventServerLabel="EventServer#bootcamp4111-region"
eventServerTarget="bootcamp.tivlab.austin.ibm.com:0"/>
      <RemoteServer aggregateStatus="PARTIAL"
hostname="dell-22.tivlab.austin.ibm.com:1918" protocol="IP.PIPE"
source="Gateway:dell-224111-GW" status="DEPLOYED"
target="Remote:REMOTE_dell-22" unix_OSAgent_installDir="/opt/IBM/ITM"
windows_OSAgent_installDir="C:\IBM\ITM">
...
```

Note that the XML results will likely contain comment attribute entries for most of the OSAgent elements indicating that they do not have agents installed. Example 5-13 shows an example.

Example 5-13 comment attribute for OSAgent element

```
..
<OSAgent comment="AMXUT2535W OS agent was not found.&#xa;&#xa;Explanation:
&#xa;The specified OS agent host name is incorrect or is not
deployed.&#xa;&#xa;Operator Response: &#xa;Verify that the OS agent host name
is correct and the OS agent is deployed before attempting the operation again."
hostname="tdw.austin.ibm.com" interp="w32-ix86" locale="US"
```

```
osAgent_installDir="" source="endpoint:tdw" status="NOT_DEPLOYED"
target="ManagedSystem:tdw"/>
...
...
```

At this stage of the upgrade, these comments for OSAgent elements are expected, as only the infrastructure has been deployed, and the comments can be ignored. Instead, the messages will be important to consider in later stages following the assessment and upgrade of endpoints.

Additional considerations

Additional items to consider when initially running the scan or rescanning to verify the settings after building the infrastructure:

- If there is already a $DBDIR/AMX/shared/analyze/scans/$TMR.xml baseline file and a new fresh rescan is attempted using **`witmscantmr -c`**, it will not be rerun. This is to prevent accidental erasure of a possibly good scan. If a rescan is required, any existing baseline file should be moved or renamed.
- A lock file is created at <wtemp>/log_witmscantmr_<timestamp>.lck (<wtemp> is the path corresponding to the output of the **`wtemp`** command and <timestamp> is in the format yyyymmdd.HH.MM.SS). This file is normally removed after the command terminates. It should be confirmed that the directory specified by wtemp still exists, and is writable by the user running the scan; otherwise there could be problems running the **`witmscantmr`** command. Because it uses a timestamp for filename uniqueness, it is unlikely to conflict with other attempts to run scans. If the scans are confirmed to be completed and these lock files still exist for some reason, they are harmless and can be removed.
- The underlying scanTMR method used for the scan requires either the super, senior, or admin role globally for the Tivoli Administrator; otherwise, an AMXUT0018E error will occur. Currently assigned roles can be confirmed using **`wgetadmin`**.
- If the command is manually terminated by the user (using Control+C or Control+Z, for example), the following message will appear:

  ```
  MXUT0021W A termination signal for this tool was received. The tool is
  busy removing temporary files and will exit upon completion..
  AMXUT0013E The command did not complete.
  ```

  The baseline XML file will not be created in this case; also, any lock file or temporary files will be removed, and the appropriate processes will be terminated.
- When the command is running, the scanTMR method implementation is initially handled by an OpmtMethods process, which eventually handles the scan using a Java class, so both OpmtMethods and a Java entry will appear

in the process list during a scan. If a scan is manually terminated for some reason, these processes should normally terminate as well. However, it may be required to manually locate and terminate the process under problematic conditions. The number and length of arguments for the Java process is long, so the class name may not appear. For reference, Example 5-14 shows a sample of Java process list output for a UNIX TMR running the scan.

Example 5-14   Sample Java entry in UNIX process list output

```
>     root 794708 884898 110 12:08:54      -  0:01
/usr/java/jre142/jre/bin/java -Xss512k
-Dbindir=/usr/local/Tivoli-4111/bin/aix4-r1
-Ddbdir=/usr/local/Tivoli-4111/bootcamp.db -Dwtemp=/tmp/
-Dtmroid=1438770036.1.348#TMF_ManagedNode::Managed_Node#
-Dregion=bootcamp4111-region -Dtmrlabel=bootcamp
-Dcwd=/usr/local/Tivoli-4111/bootcamp.db/AMX/shared/analyze/scans
-Dmmoid=1438770036.1.2193#TMF_SysAdmin::InstanceManager# -Dodispatch=4111
-Dsystemroot=NULL -Dlogid=20051019_12_08_52 -Dnlogid=0 -Dtrace=FALSE -cp
/usr/local/Tivoli-4111/bin/aix4-r1/../generic_unix/TME/ITMUpgrade/ITMUpgradeMan
ager/java/:/usr/local/Tivoli-4111/bin/aix4-r1/../generic_unix/TME/ITMUpgrade/IT
MUpgradeManager/java/java_includes/ibmjsse.jar:/usr/local/Tivoli-4111/bin/aix4-
r1/../generic_unix/TME/ITMUpgrade/ITMUpgradeManager/java/java_includes/browser.
jar:/usr/local/Tivoli-4111/bin/aix4-r1/../generic_unix/TME/ITMUpgrade/ITMUpgrad
eManager/java/java_includes/cnp_vbjorball.jar:/usr/local/Tivoli-4111/bin/aix4-r
1/../generic_unix/TME/ITMUpgrade/ITMUpgradeManager/java/java_includes/cnp.jar:/
usr/local/Tivoli-4111/bin/aix4-r1/../generic_unix/TME/ITMUpgrade/ITMUpgradeMana
ger/java/java_includes/kjrall.jar:/usr/local/Tivoli-4111/bin/aix4-r1/../generic
_unix/TME/ITMUpgrade/ITMUpgradeManager/java/java_includes/util.jar:/usr/local/T
ivoli-4111/bin/aix4-r1/../generic_unix/TME/ITMUpgrade/ITMUpgradeManager/java/ja
va_includes/ibmpkcs.jar:/usr/local/Tivoli-4111/bin/aix4-r1/../generic_unix/TME/
ITMUpgrade/ITMUpgradeManager/java/java_includes/agentconfigAPIs.jar:/usr/local/
Tivoli-4111/bin/aix4-r1/../generic_unix/TME/ITMUpgrade/ITMUpgradeManager/java/j
ava_includes/xercesImpl.jar:/usr/local/Tivoli-4111/bin/aix4-r1/../generic_unix/
TME/ITMUpgrade/ITMUpgradeManager/java/java_includes/xmlParserAPIs.jar:/usr/loca
l/Tivoli-4111/bin/aix4-r1/../generic_unix/TME/ITMUpgrade/ITMUpgradeManager/java
/java_includes/jlog.jar:/usr/local/Tivoli-4111/bin/aix4-r1/../generic_unix/TME/
ITMUpgrade/ITMUpgradeManager/java/upgradeutils.jar:/usr/local/Tivoli-4111/bin/a
ix4-r1/../generic_unix/TME/ITMUpgrad...
```

If the scan is not terminating properly, or it is suspected that underlying processes are still active even though a scan has been manually terminated, look for a Java process using the JRE path specified by **`witmjavapath`**, and that has similar -D arguments as shown above related to the scan activity. Also, the parent OpmtMethods process is the method implementation for the scanTMR method, so it could be located and terminated if there are problems.

5.3.2 Phase 2: Assess, Analyze, and Upgrade resources

With the supporting IBM Tivoli Monitoring 6.1 infrastructure deployed, the endpoints, profiles, and profile managers can be assessed in the TME environment and upgraded to corresponding situations and managed system lists in the IBM Tivoli Monitoring 6.1 environment. The Upgrade tools were designed to be flexible with options to assess or upgrade only particular resources, to allow for a phased approach with more easily identifiable checkpoints and progressive status updates. In theory, the tool used to assess the current environment could be run without arguments, in which case it would examine all profile managers that have Distributed Monitoring profiles as members, then identify the full profile manager hierarchy and endpoint subscribers for each, then identify the Distributed Monitoring profiles and monitors within them, and provide an assessment of the entire environment. Similarly, the tool used to upgrade the agents and to create corresponding situations and managed system lists could be run using a full assessment of the whole environment as a guideline. However, best practices instead involves doing a phased assessment and upgrade to provide a more manageable set of analyzable data, and to help conform with established change management procedures.

Given the flexibility of the upgrade tools, a phased assessment can still be done in many ways. Most environments use one or more structured profile manager hierarchies based on geography, platform, or system function, and these typically group similar types of endpoints. One method of phased assessment and upgrade involves identifying a particular branch of the profile manager hierarchies, then working with it to selectively upgrade particular endpoint and profile resources within it. Endpoint subscribers can be assessed and upgraded first, then profile members and monitors, then profile managers to complete the upgrade of the monitoring environment. Thus a summary of the steps would be:

- Assess a subset of endpoints.
- Upgrade the endpoints.
- Assess profiles associated with those endpoints.
- Upgrade profiles to situations.
- Assess the profile manager hierarchy.
- Upgrade the profile manager hierarchy.
- Repeat the process, iterating over the remaining profile managers, profiles, and endpoints.

The endpoints can be assessed and upgraded first so that the proper agents can be deployed and ready to handle any situations that are upgraded later. Next, selected profiles and monitors within them can be assessed and upgraded, which creates new situations; at that point, new agents would be running, and new situations would be created but not yet running, waiting to be associated with those agents. Finally, the profile managers can be assessed and upgraded; this creates new managed system lists (MSLs) which are then assigned as targets of the appropriate situations. The situations are then activated to complete the monitoring upgrade.

This allows a controlled upgrading of monitors for easier tracking progress. If instead the profile managers were assessed and upgraded directly without using a phased approach, it would automatically review all nested profile managers, profiles, and endpoints in the tree. This not only leads to a rather large set of output to review, but also requires simultaneously tracking the assessment and creation of multiple different types of entities (agents, situations, managed system lists).

The flow can be summarized in a flowchart as in Figure 5-10 on page 274.

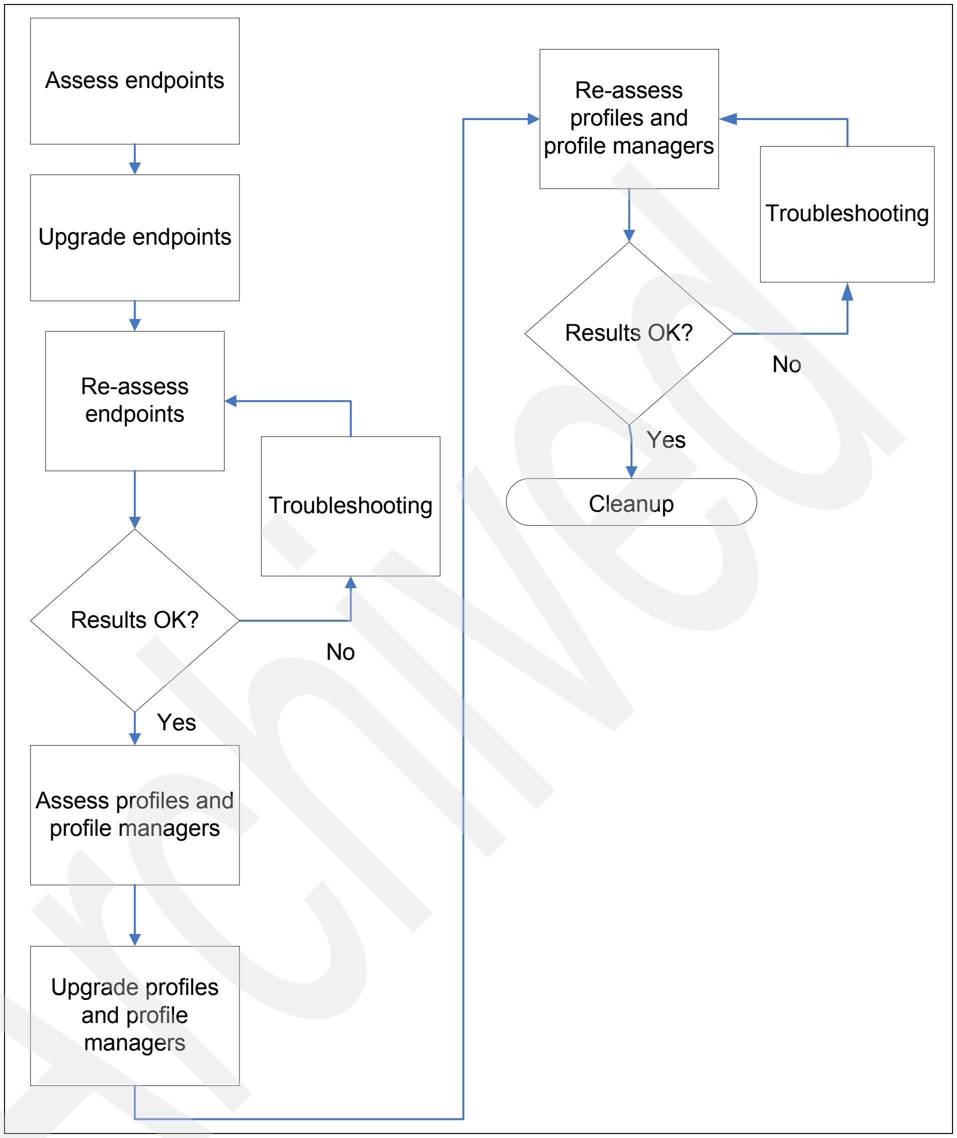


Figure 5-10 Assess, analyze, upgrade resources using phased approach

In the sample environment used for this chapter, the profile manager hierarchy with endpoint subscribers is used, as shown in Figure 5-11.

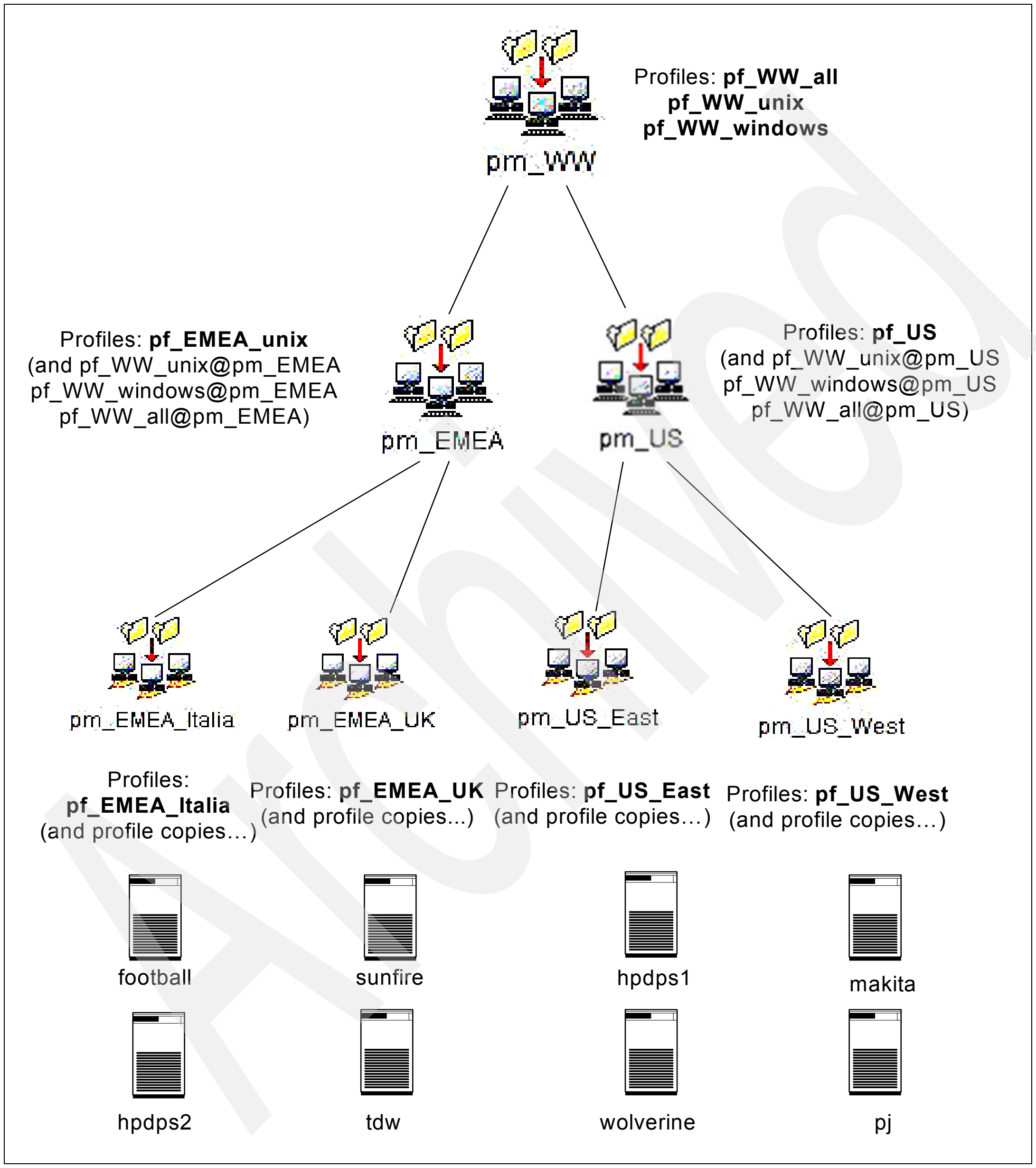


Figure 5-11 Sample environment profile manager hierarchy

Figure 5-11 on page 275 shows a top-level profile manager, pm_WW, with two subscribing profile managers: pm_EMEA and pm_US. The pm_EMEA profile manager has two nested profile managers: pm_EMEA_Italia and pm_EMEA_UK. The pm_US profile manager also has two nested profile managers: pm_US_East and pm_US_West. The endpoint subscribers for each of these lower-level profile managers is listed below the dataless profile managers.

Original profiles are listed in bold text in Figure 5-11 on page 275. Profile copies are listed for the first level, but for readability they are not listed in the lower-level profile managers.

Assess endpoints

In this sample environment, it is decided that the endpoints in the EMEA geography are considered the most important to upgrade, so an assessment is done for the four endpoints football, hdps2, sunfire, and tdw. Example 5-15 could be done, for example.

Example 5-15 Sample endpoint assessment command output

```
bootcamp# witmassess -r eplist.in -f baseline.xml
AMXUT5053I The assess tool is checking each endpoint specified for
responsiveness. This may take a while. Please wait...

AMXUT0016I The software is processing the current request.
...........
AMXUT5016I The result of the assessment is stored in the following files:
/usr/local/Tivoli-4111/bootcamp.db/AMX/shared/analyze/endpoints/sunfire.xml
/usr/local/Tivoli-4111/bootcamp.db/AMX/shared/analyze/endpoints/tdw.xml
/usr/local/Tivoli-4111/bootcamp.db/AMX/shared/analyze/endpoints/hpdps2.xml
/usr/local/Tivoli-4111/bootcamp.db/AMX/shared/analyze/endpoints/football.xml


AMXUT0008I The command completed successfully. Refer to
/usr/local/Tivoli-4111/bootcamp.db/AMX/logs/log_witmassess_20051024_15_14_24.log
for more information.
```

Because only endpoints were assessed, it creates only endpoint assessment files, one for each endpoint, in the $DBDIR/AMX/shared/analyze/endpoints directory. Example 5-16 shows a sample from one of the endpoints.

Example 5-16 Sample XML output from endpoint assessment (sunfire endpoint)

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Copyright IBM Corporation 2005-->
<endpoint aggregateStatus="INCOMPLETE" source="endpoint:sunfire" target="N/A"
xmlns="http://www.ibm.com/tivoli/itm/ep"
```

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.ibm.com/tivoli/itm/ep ep.xsd">
   <Application source="endpointClass:TMF_endpoint::endpoint"
status="INCOMPLETE" target="N/A">
      <TargetAgent product="KUX"/>
      <TargetEnvironment source="@SentryProfile:1438770036.1.2234"
target="N/A">
         <TargetEnvironmentSetting
key="ADMIN"value="Root_bootcamp4111-region"/>
         <TargetEnvironmentSetting key="ENDPOINT_CLASS" value="ENDPOINT"/>
         <TargetEnvironmentSetting key="LCF_DATDIR"
value="/space/Tivoli/lcf2/dat/1"/>
         <TargetEnvironmentSetting key="HOST_FQDN"
value="moonwater.tivlab.austin.ibm.com"/>
         <TargetEnvironmentSetting key="NLSPATH"
value="/space/Tivoli/lcf2/generic/msg_cat/%L/%N.cat:/space/Tivoli/lcf2/generic/
msg_cat/%l/%N.cat:/space/Tivoli/lcf2/generic/msg_cat/C/%N.cat"/>
         <TargetEnvironmentSetting key="TISDIR"
value="/space/Tivoli/lcf2/dat/1"/>
         <TargetEnvironmentSetting key="ENDPOINT" value="sunfire"/>
         <TargetEnvironmentSetting key="ENDPOINT_OID" value="1438770036.6.0+"/>
         <TargetEnvironmentSetting key="LCF_LIBDIR"
value="/space/Tivoli/lcf2/lib/solaris2"/>
         <TargetEnvironmentSetting key="OPERATOR"
value="root@bootcamp.tivlab.austin.ibm.com"/>
         <TargetEnvironmentSetting key="HOST_OID" value="1438770036.6.0+"/>
         <TargetEnvironmentSetting key="PROFILEOID"
value="1438770036.1.2234#Sentry::All#"/>
         <TargetEnvironmentSetting key="INTERP" value="solaris2"/>
         <TargetEnvironmentSetting key="LCFROOT" value="/space/Tivoli/lcf2"/>
...(other TargetEnvironmentSetting values omitted)...
</TargetEnvironment>
...(other TargetEnvironment entries skipped)...
```

This is a summary of the activity that occurs for each particular endpoint, to generate this file:

- The OpmtMethods process is again started on the TMR. Connectivity to each endpoint being assessed is tested by doing a "browse" downcall. It does the equivalent of a **`wadminep <endpoint> view_config_info downcall`** to test whether the endpoint is responsive. If it is not, it will log a AMXUT5052W warning and continue to the next endpoint. If it is responsive, it continues and spawns the JRE to load the com.ibm.opmt.asess.Assess class and continue the assessment.
- An EPA_GetDataStore downcall is done to the endpoint to decrypt and read some of the contents of the DM engine persistent store ($LCF_DATDIR/.sntcfg/data). The file contains details regarding which

profiles were last distributed and running in the engine, along with environment-related context variables associated with each profile. Acquiring the list of profiles here enables it to pinpoint profiles that had actually been used by the engine. This data eventually will be stored in $DBDIR/AMX/data/environments and used later for monitors deemed to be “custom” and that will require the environment context information when deployed to the Universal Agent. Note that it is directly analyzing the files, so although the endpoint (lcfd) must be available to handle the downcalls, the Distributed Monitoring engine (dm_ep_engine) does not have to be running, doing the downcall does not restart the engine if it is stopped, and it does not contact the engine process in any way.

- It confirms the generic interp-type (Windows, UNIX, or Linux) to determine what corresponding OS Agent should be used later.
- An EPA_verifyOSAgent downcall is done, which further verifies that the OS type and version are supported. It checks for:
  - Windows: at least Windows 2000 (verified using a GetVersionEx() call).
  - Solaris: at least 5.8 or higher in uname output.
  - AIX: at least 5.x.
  - HP-UX: at least 11.x.
  - Linux: at least 2.4 kernel or higher.
- The same downcall then confirms whether the Agent is already installed by scanning for installation tag files. On Windows, it looks in the <installDir>\INSTALLITM\VER directory for the KNTWICMA.ver file. For UNIX, it looks for any <installDir>/registry/ux*.ver files. For Linux, it looks for any <installDir>/registry/lz*.ver files. If it confirms that there is a tag file in the chosen installation directory, it assumes that the Agent is already installed.
- The downcall finishes by confirming available disk space using the following estimates for required installation space:
  - Windows: 85 MB
  - Solaris:155 MB
  - AIX: 210 MB
  - HP-UX: 175 MB
  - Linux: 230 MB

Note: After problems were encountered with some endpoint upgrades for our lab, it was noticed that the sizes being checked above seem to consist mostly of the size of an agent_base_install.tar dependency file that is initially pushed. This file has to be temporarily extracted from the cache on the endpoint. It extracts it to the directory where the agent_base_install.tar file was pushed. Later, the installation is done to the directory specified for the new Agent installation directory. Thus, the actual space required for an installation is two times the size above on the filesystem where the endpoint is currently installed, plus at least the equivalent size on the filesystem where the Agent is being installed.

After the upgrade, the files under the cache on the endpoint are removed, so that space is only temporarily used. Thus, the value for the sizes above is an approximation of the initial size to expect for the new Agent directory that is created.

For example: An AIX endpoint is upgraded.

- Approximate space required for endpoint directory: 210 x 2 = 420 MB
- Approximate space required for Agent install directory: 210 MB
- Approximate total space required to do install: 420 + 210 = 630 MB
- Approximate total space used after temporary files removed = 210 MB

The actual space used by the new Agent is likely smaller than the original tar file, but the estimate is a good starting point to allow for growth in any historical data files and logs being generated later.

- Temporary files named assess_transfer_<timestamp> are created in the wtemp directory but are removed if debug tracing is not enabled.

  Assuming that these checks all work fine, the output in the $ENDPOINT.xml file shows content similar to the figure above.

Upgrade endpoints

At this stage, the endpoints can be upgraded. Before outlining the behavior, it is important to note at this stage that if the upgrade tool **`witmupgrade`** is used to handle the upgrade of the endpoint to an IBM Tivoli Monitoring 6.1 OS Agent and Universal Agent, that it will use the dependency services provided by Framework to distribute the necessary files and handle the silent background install. At LAN speed for a small number of endpoints such as this, this may not be a problem. In larger environments distributed across networks utilizing different link speeds, this could become a performance issue.

Fortunately, the upgrade tools are not the only means for installing an OS Agent. The assessment and upgrade are being done here using a phased approach, which enables a user to uncouple the actual endpoint upgrade process from the rest of the assessment and upgrade. There are multiple ways to install an OS Agent, including:

- Agent Deployment functionality: IBM Tivoli Monitoring 6.1 adds Agent Deployment functionality via the **tacmd createNode** command. This allows initial deployment of the base OS Agent for a system. Following this, the **tacmd addSystem** command could be used to deploy other non-OS Agents to the system.
- Tivoli Configuration Manager: Software packages can be built for the Agents and distributed using Configuration Manager; this would use the services of MDist2 in Framework to more easily manage the distribution of a larger number of Agents.
- Manual installation: The Agents can always be installed manually using a command line for UNIX and Linux and via GUI or silent install for Windows.

In Example 5-17, just one endpoint, sunfire, is upgraded to show sample output. At LAN speeds, this particular example required about two minutes to complete.

Example 5-17 Sample output

```
bootcamp# witmupgrade -x sunfire.xml -f ../scans/baseline.xml -u

AMXUT0016I The software is processing the current request.
........................................................


AMXUT7700I The result of the command is stored in the following files:
/usr/local/Tivoli-4111/bootcamp.db/AMX/shared/analyze/endpoints/upgrade/sunfire
_20051025_11_33_39.xml



AMXUT0008I The command completed successfully. Refer to
/usr/local/Tivoli-4111/bootcamp.db/AMX/logs/log_witmupgrade_20051025_11_33_39.l
og for more information.
```

This is a summary of the activity during the upgrade:

- A migrate method is called, which calls an installOSAgent method to initiate the upgrade. (This activity is visible in the odstat and wtrace -jHk $DBDIR output on the TMR server.) An EPA_verifyOSAgent downcall is done again to the target endpoint to reconfirm that the OS and disk space are still suitable and that an Agent still should be installed; and a browse method is done again to confirm endpoint availability. This may seem repetitive, but it has to do this again because it is possible that, since the time of the assessment, the conditions could have changed.

- A set_config downcall is done to the endpoint, which changes the cache_limit endpoint configuration setting to 250 MB (250000000). This is to allow for additional space to deploy and install the Agent.
- An EPA_installOSAgent downcall occurs. This is associated with the ITMAgentDeploy dependency set, which pushes a large agent_base_install.tar image and silent install files. The tar file is temporarily located under the $LCF_DATDIR file cache hierarchy, so it can be extracted into the Agent installation directory. Remember that the baseline.xml file contains the OSAgent settings for the endpoint to determine which directory to install into, and which Remote TEMS to connect to. If there were custom settings here, they would be used; otherwise, the defaults in the RemoteServer element would be used. After extraction, it uses silent install files to install the GSKit, OS Agent, and Universal Agent, and starts the Agents. Afterward, it removes the temporary installation files that were under $LCF_BINDIR/TME/ITMUpgrade.
- Another EPA_verifyOSAgent downcall occurs just to verify that the OS Agent is installed; it looks for the tag files mentioned earlier to confirm.
- The cache_limit is set back to the original value via a set_config downcall.
- A $DBDIR/AMX/data/status.properties file is updated for each endpoint to show whether it was upgraded successfully. This is used later to determine whether it was able to generate custom environment files under $DBDIR/AMX/data/environment for the endpoints. Any content in that environment subdirectory will be used if custom monitors have to be mapped to custom situations later during the profile upgrades.

This example environment used the Solaris OS, so the new installation directory can be accessed to confirm that the Agents are present and running by switching to <installdir>/bin and running the ./cinfo utility (or alternatively, by examining **ps -ef** output). Example 5-18 shows this for the endpoint sunfire on the host moonwater.

Example 5-18 Using cinfo or ps -ef to confirm UNIX / Linux Agent status

```
moonwater:/space/IBM/ITM/bin# ./cinfo

*********** Tue Oct 25 15:12:36 CDT 2005 ******************
User      : root         Group: other root bin sys adm uucp mail tty lp nuucp
daemon
Host name : moonwater    Installer Lvl: 400 / 100
CandleHome: /space/IBM/ITM
***********************************************************


    -- CINFO Menu --
 1) Show products installed in this CandleHome
 2) Show which products are currently running
 3) Show configuration settings
 4) Show installed CD release versions
 X) Exit CINFO
2

*********** Tue Oct 25 15:12:37 CDT 2005 *****************
User      : root         Group: other root bin sys adm uucp mail tty lp nuucp
daemon
Host name : moonwater    Installer Lvl: 400 / 100
CandleHome: /space/IBM/ITM
***********************************************************
Host     Prod    PID     Owner   Start    ID  ..Status
moonwater        ux      10887   root    11:13:35 None ..running
moonwater        um      11196   root    11:13:43 None ..running

    -- CINFO Menu --
 1) Show products installed in this CandleHome
 2) Show which products are currently running
 3) Show configuration settings
 4) Show installed CD release versions
 X) Exit CINFO
x
moonwater:/space/IBM/ITM/bin# ps -ef| egrep "kux|kum"
    root 14148 10023  0 15:12:52 pts/1    0:00 egrep kux|kum
    root 11196     1  0 11:13:43 ?        0:01
/space/IBM/ITM/sol286/um/bin/kuma610
    root 10896 10895  0 11:13:35 ?        0:00 kuxdstat 1
    root 10895 10890  0 11:13:35 ?        0:00 sh -c kuxdstat 1
    root 10887     1  0 11:13:33 ?        0:04
/space/IBM/ITM/sol286/ux/bin/kuxagent
    root 10891 10890  0 11:13:35 ?        0:00 kux_vmstat 30 12
```

Alternatively for Windows, the Task Manager could be accessed to confirm the status. Also, the Tivoli Enterprise Portal client could be used to confirm that the Agents were able to access their assigned Remote TEMS and to appear as updated nodes in the GUI.

Reassess endpoints

After doing a large number of endpoints, to confirm the overall status of which endpoints have been upgraded, a reassessment can be done. For example:

```
witmassess -e eplist.in -f baseline.xml
```

`eplist.in` is a filename containing the endpoint resources used earlier (@endpoint:football, @endpoint:sunfire, and so on). It reassesses the current conditions, writing new versions of <endpoint>.xml files to the $DBDIR/AMX/shared/analyze/endpoints directory. If the Agent was deployed and started OK, the aggregateStatus shows `COMPLETE` in each XML. For example, for the Windows endpoint tdw, the tdw.xml output shows Example 5-19.

Example 5-19 Updated endpoint XML after reassessment

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Copyright IBM Corporation 2005-->
<endpoint aggregateStatus="COMPLETE" source="endpoint:tdw" target="N/A"
xmlns="http://www.ibm.com/tivoli/itm/ep"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.ibm.com/tivoli/itm/ep ep.xsd">
  <Application source="endpointClass:TMF_endpoint::endpoint" status="COMPLETE"
target="N/A">
      <TargetAgent product="KNT"/>
...
```

Alternatively, remember that to get overall status, another **`witmscantmr`** can be done to get an updated baseline_<timestamp>.xml file:

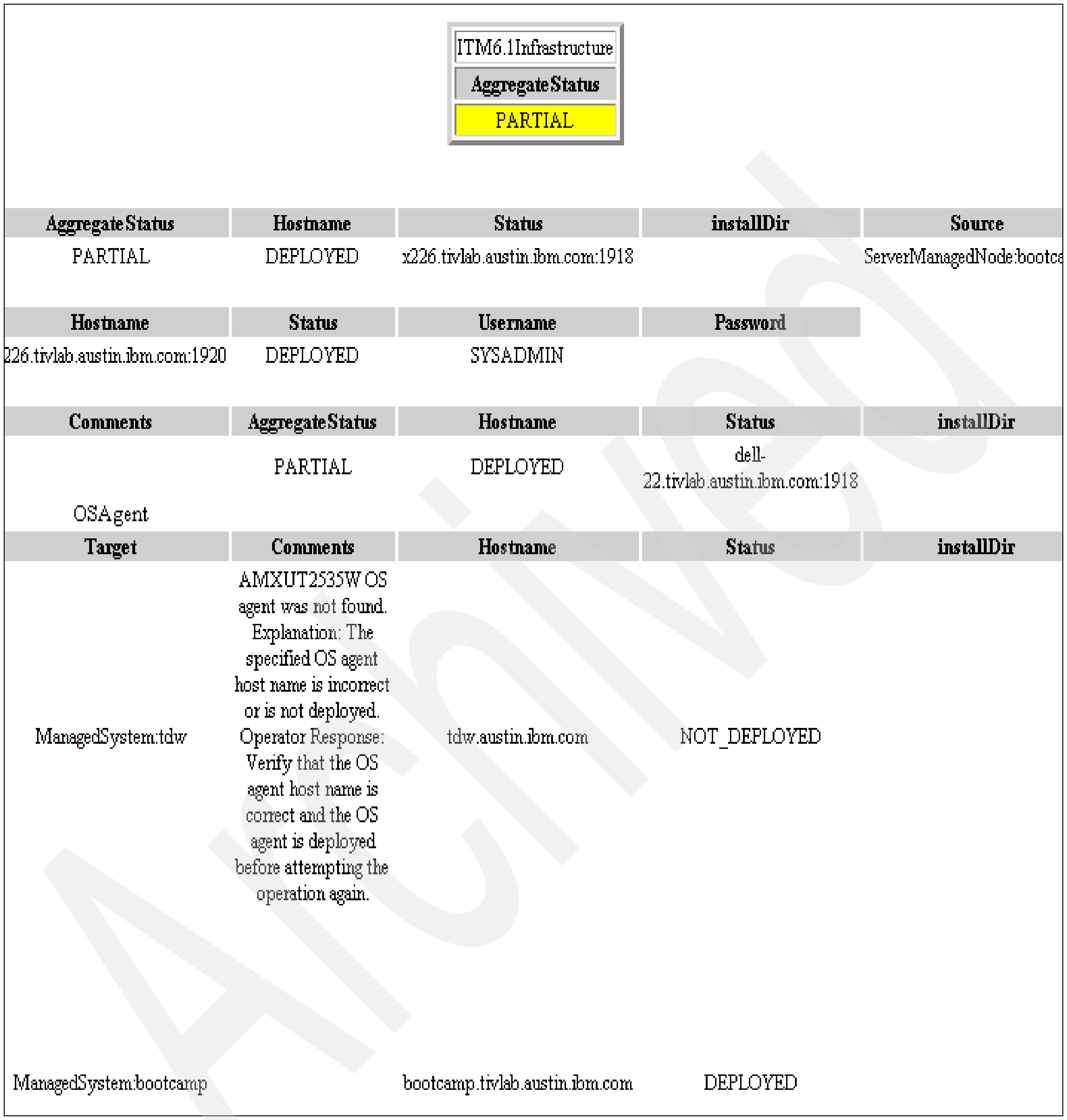
```
witmscantmr -f baseline.xml -v
```

This shows overall status in XML format; Example 5-20 shows an excerpt with the new status for deployed endpoints.

Example 5-20 New baseline_<timestamp>.xml output showing updated Agent status

```
...
...
<OSAgent hostname="pj.austin.ibm.com" interp="w32-ix86" locale="US"
osAgent_installDir="" source="endpoint:pj" status="DEPLOYED"
target="ManagedSystem:pj"/>
...
...
```

Again, the XML can be formatted using the tmr.xsl stylesheet. (Updated stylesheets are planned for the final release to deal with formatting problems seen in Figure 5-12 on page 284.)



ITM6.1Infrastructure
AggregateStatus
PARTIAL

AggregateStatus	Hostname	Status	installDir	Source
PARTIAL	DEPLOYED	x226.tivlab.austin.ibm.com:1918		ServerManagedNode:bootca

Hostname	Status	Username	Password
226.tivlab.austin.ibm.com:1920	DEPLOYED	SYSADMIN	

Comments	AggregateStatus	Hostname	Status	installDir
OSAgent	PARTIAL	DEPLOYED	dell-22.tivlab.austin.ibm.com:1918	

Target	Comments	Hostname	Status	installDir
ManagedSystem:tdw	AMXUT2535W OS agent was not found. Explanation: The specified OS agent host name is incorrect or is not deployed. Operator Response: Verify that the OS agent host name is correct and the OS agent is deployed before attempting the operation again.	tdw.austin.ibm.com	NOT_DEPLOYED	
ManagedSystem:bootcamp		bootcamp.tivlab.austin.ibm.com	DEPLOYED	

Figure 5-12 tmr.xsl stylesheet

When troubleshooting endpoints that show NOT_DEPLOYED status:

- Review messages appearing in $DBDIR/AMX/logs/log_witmupgrade_<timestamp>.log. If debug tracing was enabled, further details appear in

$DBDIR/AMX/trace/trace_witmupgrade_<timstamp>.log. If there are errors mentioning that the endpoint was not reachable, confirm that the endpoint is reachable and responsive now. (**`wadminep <endpoint> view_config_info`** is a simple test.)

- If the endpoint was responsive, review the endpoint itself to confirm what happened during the deployment phase. If the log level was not 3 for the lcfd.log and not enough logging content was shown to indicate what failed, use **`wep <endpoint> set_config log_threshold=3`** to dynamically update the log level, and retry the upgrade for the affected endpoint; if the problem happens again, the new lcfd.log will contain additional status messages when downcalls were being handled. Look for messages indicating an unsupported platform or version, or a lack of available disk space. If there was a lack of disk space on a second attempt after an initial error, confirm that there is not an agent_base_install.tar still there. In testing here, occasionally this file was not removed by an initial upgrade failure, so previous upgrade attempts afterward failed with an error mentioning a lack of available space.
- If the Agent was deployed successfully but there were problems later, confirm what happened in the installation phase and Agent startup phase. The silent install logs status messages to lcfd.log, but additional content would exist under the new Agent installation directory, in a $CANDLEHOME/logs directory, when attempting to configure or start the Agent.

Mapping Distributed Monitoring resources to IBM Tivoli Monitoring 6.1

Before moving to the next phase of assessment and upgrade, the details of what is being assessed and upgraded from the profiles and profile managers must be discussed. This serves as a basis for a best-practices discussion and example of profile and profile manager assessment and upgrade.

Figure 5-13 on page 286 summarizes the structure of a Distributed Monitoring profile.

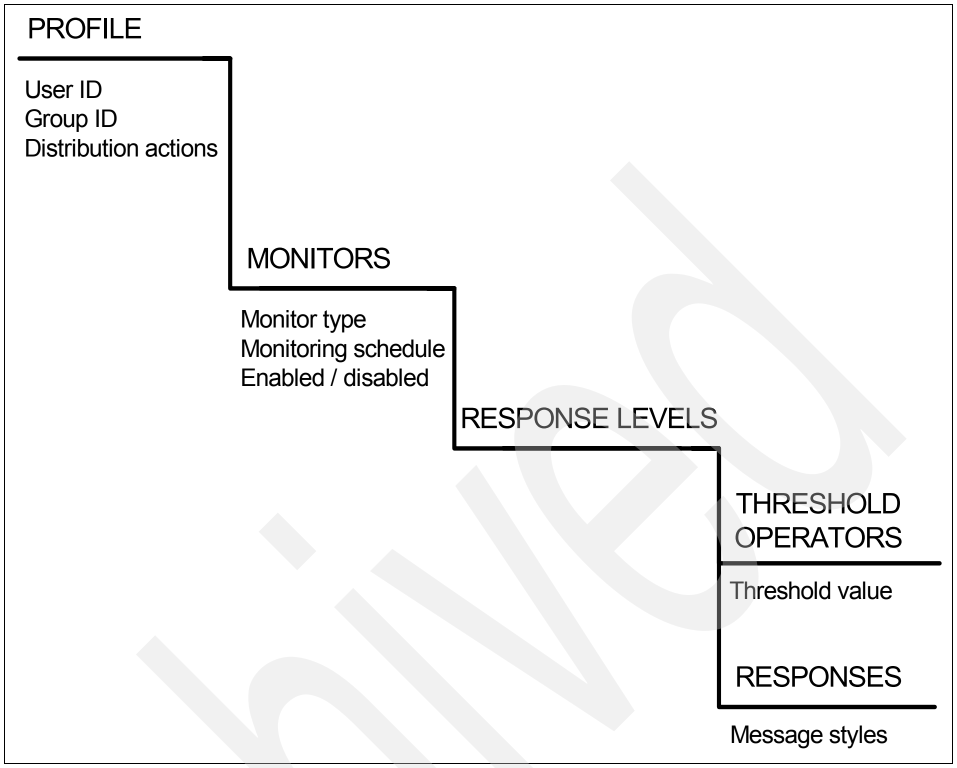


Figure 5-13 Simplified Distributed Monitoring profile structure

For each of the items above, a description of what is done for IBM Tivoli Monitoring 6.1 follow.

Profile-level settings

The following settings cover the profile and all of its monitors as a whole:

- Profile User ID and Group ID

  The user ID and group ID would be used in the following cases for Distributed Monitoring:

  – Running custom monitors (such as numeric script or string script monitors).

  – Running “Run program” responses.

  – Using the “Log to file” response.

  – Running the “task” response (it overrides the task’s embedded “user” setting).

  – Running custom monitors from custom-built collections where the SetID=YES directive is used.

These user ID and group ID settings are not mapped to situations in IBM Tivoli Monitoring 6.1. This could result in problems for those elements that are mapped to situations, such as custom monitors and “run program” responses. Some observations to consider:

  - For Distributed Monitoring custom monitors that were mapped to V6.1 situations for a Universal Agent, and Distributed Monitoring “run program” responses that were mapped to 6.1 actions, it is possible that the desired user ID and group ID configured in the monitoring profile for the monitor may not map to the user and group used for the corresponding situation in the Universal Agent. If the custom monitors and “run program” responses required a particular user due to permissions and environment variables, it could lead to problems running the mapped situations in the new Agents. Consider this if problems occur when running the corresponding upgraded situations and actions.
  - Additionally, ID maps (sometimes referred to as widmap settings because of the **widmap** command) were sometimes used for profile user ID and group ID settings (especially the frequently customized $root_user and $sentry_user ID maps). ID maps are not used in IBM Tivoli Monitoring 6.1, so if custom monitors and “run program” responses are expecting these IDs, the corresponding values should be confirmed using **widmap list_entries <ID_Name>**.
  - Finally, any custom scripts or “run program” responses using relative paths instead of fully qualified paths should be reviewed, as the current working directory most likely will be different in upgraded situations and actions than it was for the old monitors and responses. Custom script monitors and “run program” responses may have been written originally with the assumption of $LCF_DATDIR as the working directory when using paths to files or commands.

- Distribution Actions: These are not mapped to 6.1. For the case of endpoints that had already had the profile distributed in the past, this is not normally an issue, as the appropriate actions would already have been completed. The only typical concern involves distribution actions that would have been used to push new scripts to any newly added endpoints in the future. Other means must be explored for having these files pushed to new agent targets in the future.

Monitors

The type of monitor must be considered, along with its monitoring cycle time and enabled/disabled state:

- Monitor type

  When considering whether monitors will be mapped to corresponding situations, there are three main categories: unsupported / deprecated monitors, monitors mapped to situations in OS Agents, and monitors mapped to situations in the Universal Agent:

- *Unsupported / deprecated monitors*: A list of supported and unsupported monitors for the upgrade are included in HTML format on the base installation media for the Distributed Monitoring Upgrade tools, in the BASE/UPGINFO subdirectory. If the installation media is not accessible, then a list of the unsupported collections and monitors that is actually loaded by the upgrade tools during assessment also exists in the $BINDIR/../generic_unix/TME/ITMUpgrade/ITMUpgradeManager/java/data/unsupported.properties file. The format used in this file is coll_<MonitoringCollection>.[monitor].solution=AMXUT7508W, where <MonitoringCollection> is the monitoring collection name, and [monitor] is an optional monitor name. A monitor listed in this file is not mapped to a new situation.
- *Monitors mapped to situations in OS Agents*: Numerous monitors from the Universal, Unix_Sentry, and w2k_* Windows monitoring collections are automatically mapped to corresponding situations in the OS Agents. A set of mapping files are loaded and used by the assessment tools to determine which monitors are mapped to new OS Agent situations. The XML mapping files are located in the $DBDIR/AMX/data/mapping directory; there's one for each target OS Agent. The subdirectory $BINDIR/../generic_unix/ITMUpgrade/ITMUpgradeManager/MappingFiles also has original copies. The files can be reviewed to understand which monitors from particular collections are mapped to new situations.
- *Monitors mapped to situations in the Universal Agent*: If the monitor is not listed as unsupported or deprecated, and still does not match one of the mapped entries above, then it is considered custom. This includes monitors such as custom numeric / string scripts and asynchronous monitors, but also includes monitors from custom-built collections.

In the example assessment and upgrade output that is done later in the chapter, more detail will be provided to explain what situations are created, how the situation names are derived, what managed system lists are created, and how those MSL names are derived.

- Monitoring schedule

  Only the monitor cycle time is mapped to the IBM Tivoli Monitoring 6.1 Sampling Interval; the monitoring schedule controlling hours and days is not mapped. This can be a concern for monitors set up with customized schedules, such as when maintenance windows are used and monitors are disabled during certain time periods. For these types of custom schedules, IBM Tivoli Monitoring 6.1 includes several implementation possibilities:

– Local_Time attribute group: The attributes in this attribute group include the day and time at the Agent site; these attributes can be added to the situation formula for situations requiring maintenance windows.

– Universal_Time attribute group: These attributes include the day and time at the TEMS to which the Agent is connected.

– All Managed Systems predefined situations: These handle day-of-week and time-of-day attributes. These can be used as embedded situations within the situation formula of situations that require customized schedules. In the Situation Editor, these appear under the section for All Managed Systems and have descriptive names such as PrimeShift and NonPrimeShift.

Notes:

- An IBM Tivoli Monitoring 6.1 situation interval can be as low as one second. Distributed Monitoring V3.7 was limited to the lowest of one minute.
- The Distributed Monitoring V3.7 schedules do not get transferred over during the upgrade process. Only the monitoring frequencies are retained.

- Enabled/disabled setting

  If the monitor is enabled in the profile, then it will map to the “Run at startup” setting being enabled in the corresponding mapped situations.

Response levels

WIth Distributed Monitoring, a single monitor could have multiple response levels, with each response level having its own set of responses. Because IBM Tivoli Monitoring 6.1 does not use multiple response levels within a single situation, the solution for upgrade is to map each response level in a Distributed Monitoring monitor to separate situations in 6.1, and then use a naming convention that makes each one discernible.

For example, for a “filesystem percent space used” monitor (diskusedpct) that had Critical and Severe response levels with responses set up, it would map to two situations: a situation with a name like diskusedpct_critical and a second situation with a name like diskusedpct_warning; each would then have its own separate responses set up within them. The actual naming convention used is more complex and will be outlined in the examples that follow later in the chapter.

Because there are more Distributed Monitoring default response levels than there are 6.1 situation event severities, they must be mapped down to fit the three available severities of Critical, Warning, or Informational. The default response levels map as shown in Table 5-4.

Table 5-4 Response level-to-situation severity mapping table

Response level	Precedence level	Mapped ITM 6.1 severity
critical	1000	Critical
severe	500	Critical
warning	100	Warning
normal	0	Informational
always	-1	Informational
E.EXEC	-1000	Warning

Note that the precedence level that corresponds to a response level is viewable using the Distributed Monitoring `wlslevels <ProfileName>` command. This becomes particularly important when custom response levels are used in profiles. In those cases, the custom levels are mapped using the precedence level as in Table 5-5.

Table 5-5 Precedence level-to-situation severity mapping table

Precedence level	Mapped ITM 6.1 severity
500 or higher	Critical
100 - 499	Warning
0 - 99	Informational
Less than zero	Warning

For any monitor being upgraded, since a single monitor with multiple thresholds is being mapped to multiple situations, the situation formulas for each mapped situation must be updated to account for the fact that they are all being calculated separately. Recall that when a single Distributed Monitoring monitor runs with multiple response levels, the algorithm for determining what response level to use is as in Figure 5-14 on page 291.

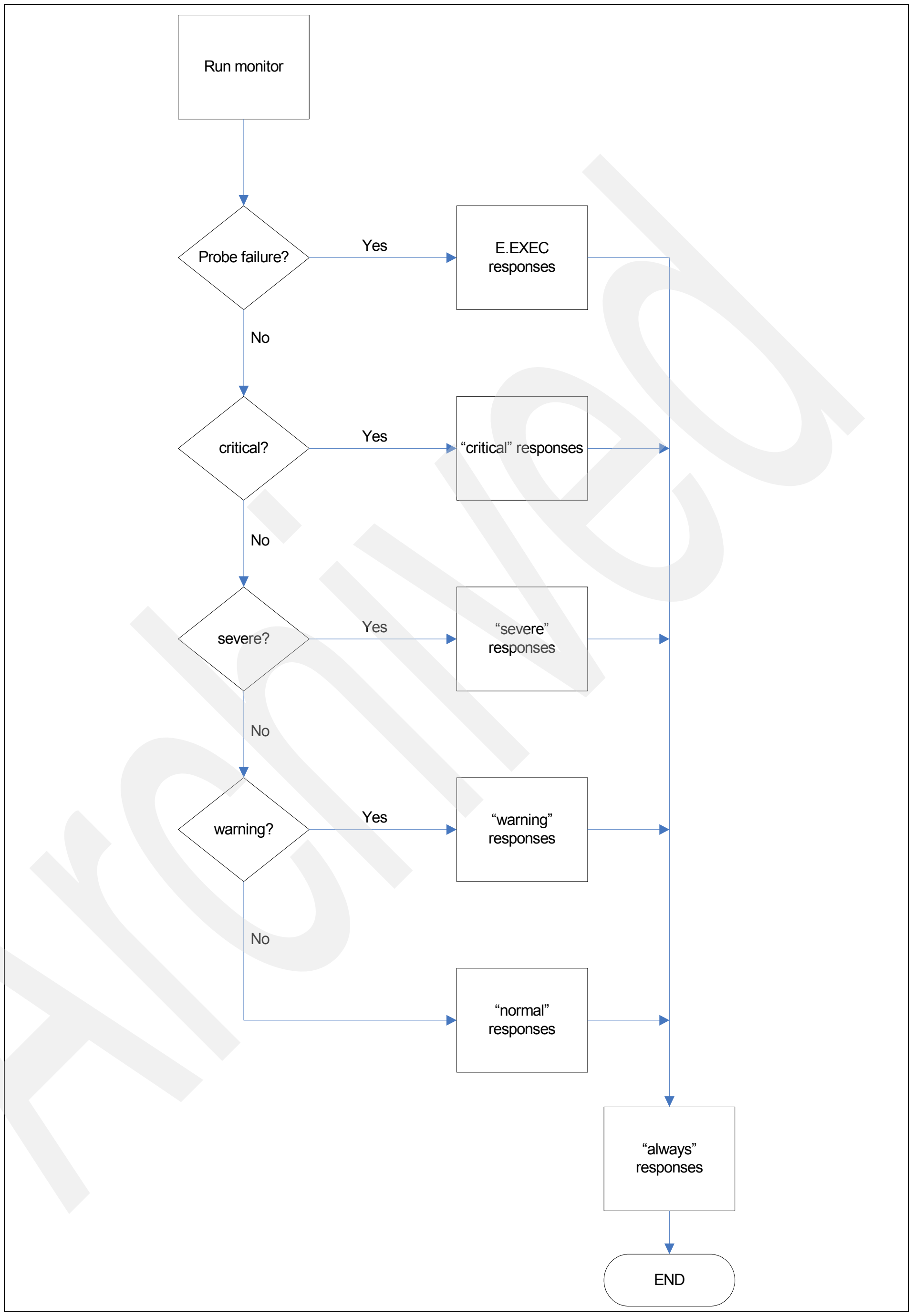


Figure 5-14 Distributed Monitoring monitor response level determination flowchart

The flow above ensures that only one response level ever matches (besides the "always" response level, because the responses defined there always run, even when there is a failure). If multiple response levels are mapped to multiple separate situations, and each has a separate situation formula, then the formulas used must account for threshold values to ensure that only one of the situations from the critical, severe, warning, normal, or E.EXEC levels actually fires. For example, if a Percent Disk Space Used monitor (diskusedpct) has response levels for critical if diskusedpct < 5, and warning if diskusedpct < 10, then two situations would have to be created using formulas like this:

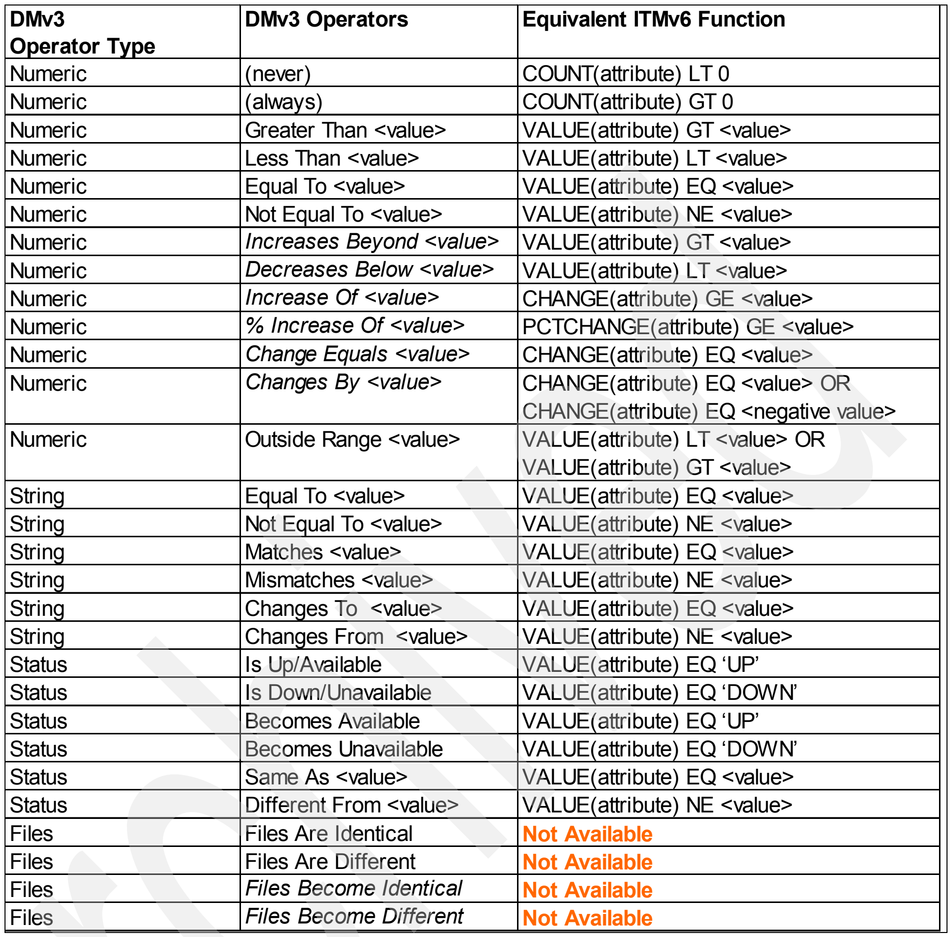
```
situation_critical: if diskusedpct < 5
situation_warning: if diskusedpct < 10 AND diskusedpct > 5
```

The Warning situation had to have the formula extended to ensure that both situations would not be considered true. If it simply used `if diskusedpct < 10`, then both the situation_warning and the situation_critical would be true, and that is not desired. The assessment tool handles this type of situation formula mapping automatically when multiple response levels are used.

Note: With the versions of code used for this book, the E.EXEC response level was not being mapped to situations. This may have been changed intentionally, because E.EXEC is a special level for Distributed Monitoring that is used only for probe failures. and IBM Tivoli Monitoring 6.1 does not have corresponding functionality for mapping situations to this type of condition. One theoretical alternative is to create a situation to track the overall situation states for a managed system, and simply alert when any of them were in a Problem state; but this is outside the scope of the upgrade tool.

Threshold operators

Distributed Monitoring used a variety of numeric and string-based threshold operators, most of which are mapped to corresponding operators used in situation formulas for IBM Tivoli Monitoring 6.1. Figure 5-15 on page 293 summarizes the current mapping.



DMv3 Operator Type	DMv3 Operators	Equivalent ITMv6 Function
Numeric	(never)	COUNT(attribute) LT 0
Numeric	(always)	COUNT(attribute) GT 0
Numeric	Greater Than <value>	VALUE(attribute) GT <value>
Numeric	Less Than <value>	VALUE(attribute) LT <value>
Numeric	Equal To <value>	VALUE(attribute) EQ <value>
Numeric	Not Equal To <value>	VALUE(attribute) NE <value>
Numeric	*Increases Beyond <value>*	VALUE(attribute) GT <value>
Numeric	*Decreases Below <value>*	VALUE(attribute) LT <value>
Numeric	*Increase Of <value>*	CHANGE(attribute) GE <value>
Numeric	*% Increase Of <value>*	PCTCHANGE(attribute) GE <value>
Numeric	*Change Equals <value>*	CHANGE(attribute) EQ <value>
Numeric	*Changes By <value>*	CHANGE(attribute) EQ <value> OR CHANGE(attribute) EQ <negative value>
Numeric	Outside Range <value>	VALUE(attribute) LT <value> OR VALUE(attribute) GT <value>
String	Equal To <value>	VALUE(attribute) EQ <value>
String	Not Equal To <value>	VALUE(attribute) NE <value>
String	Matches <value>	VALUE(attribute) EQ <value>
String	Mismatches <value>	VALUE(attribute) NE <value>
String	Changes To <value>	VALUE(attribute) EQ <value>
String	Changes From <value>	VALUE(attribute) NE <value>
Status	Is Up/Available	VALUE(attribute) EQ 'UP'
Status	Is Down/Unavailable	VALUE(attribute) EQ 'DOWN'
Status	Becomes Available	VALUE(attribute) EQ 'UP'
Status	Becomes Unavailable	VALUE(attribute) EQ 'DOWN'
Status	Same As <value>	VALUE(attribute) EQ <value>
Status	Different From <value>	VALUE(attribute) NE <value>
Files	Files Are Identical	**Not Available**
Files	Files Are Different	**Not Available**
Files	*Files Become Identical*	**Not Available**
Files	*Files Become Different*	**Not Available**

Figure 5-15 Formula mapping

Responses

Some Distributed Monitoring responses are mapped directly to corresponding functionality in IBM Tivoli Monitoring 6.1; others have either implicit mappings or are not mapped. For these cases, alternatives are mentioned where available.

- TEC Event

  This is by far the most commonly used response, and much was done to handle upgradability:

  – To determine the TEC Event class name: The TEC Event Class name of a situation that was upgraded from a Distributed Monitoring monitor/response level should match the original class names that were used in Distributed Monitoring. It was structured this way to prevent a user from having to analyze and redo multiple rules in rulebases to handle class name changes. If TEC Synchronization is enabled on the Hub TEMS, then situation events are forwarded to an IBM Tivoli Enterprise Console Server. It loads a series of *.map mapping files in the $CANDLEHOME/tables/<HubTEMSName>/TECLIB directory, and examines the names of the situations that are being forwarded. It matches the first portion of the situation name to an entry in one of the mapping tables to determine what appropriate class name to use. For example, if a Distributed Monitoring diskusedpct monitor has a critical response level setup that gets mapped to a situation named UX_USDkUPct_12340_critical, it finds the entry in the mapping file $CANDLEHOME/tables/<HubTEMSName>/TECLIB/kux.map that has:

```
<situation name="UX_USDkUPct*">
            <class name="Sentry2_0_diskusedpct"/>
```

Thus, it uses the regular expression UX_USDkUPct* to match to the situation name, and it maps to the same Sentry2_0_diskusedpct class that it used in Distributed Monitoring. Note that this means that when editing assessment XML output, the first portion of the situation name should not be edited, to avoid potential problems mapping to the correct class name. Situation names that do not have mapped equivalents use the ITM_Generic class name.

– To determine the TEC Event severity: This depends on the source monitor/response level that was upgraded.

  • If the new situation comes from a monitor's response level that had an IBM Tivoli Enterprise Console response set up within it, then the original severity of that IBM Tivoli Enterprise Console response is preserved and used. For example, if a monitor has a critical response level that had a TEC Event response set to send a TEC Event at a MINOR level, then if the new situation that was upgraded became true, and TEC Synchronization was being used, it would forward a TEC Event with the original MINOR event level that was used in Distributed Monitoring. It is able to do this because the original severity information was preserved during the assessment and upgrade of the monitors and responses; in the assessment XML output, an attribute called eventSeverity is used to maintain this information so that when upgraded, the setting can be kept. During **witmupgrade**, this information is updated in the Hub TEMS file $CANDLEHOME/tables/<HubTEMSName>/TECLIB/tecserver.txt file. It does this via an EPA_setEventForwarding downcall to the endpoint that was set up earlier on the Hub TEMS. This tecserver.txt file can be used to map situations to specific IBM Tivoli Enterprise Console Servers and to specific severities. Chapter 10, "Integration with IBM

Tivoli Enterprise Console" on page 583 explains the tecserver.txt file in more detail.

- If the new situation comes from a monitor's response level that did not have any IBM Tivoli Enterprise Console responses set up within it, and if there are no customizations in the tecserver.txt file on the Hub TEMS for this situation type, then the forwarded TEC Event severity is based on the ITM 6.1 severity of the situation. Because there are only three IBM Tivoli Monitoring 6.1 severities, the mapping is simple (Table 5-6).

Table 5-6 ITM 6.1 severities mapped to forwarded TEC Event severities

ITM 6.1 situation severity	**Forwarded TEC Event severity**
Critical	Critical
Warning	Minor
Informational	Warning

– To determine the TEC Event server: Again, this depends on the configuration of the source monitor.

- If the situation was upgraded from a monitor with a response level that had the IBM Tivoli Enterprise Console response set up within it, it will preserve the IBM Tivoli Enterprise Console Server used. It can do this because the assessment mapped the Event server name to an actual host name and port, and stored that information in an eventServer attribute in the XML. Later during the upgrade, the information is forwarded in the same EPA_setEventForwarding downcall to the Hub TEMS tecserver.txt file.
- If the situation was upgraded from a monitor with a response level that did not have a TEC Event response setup, it will use the default IBM Tivoli Enterprise Console Server configured for the Hub TEMS. This information is stored in the om_tec.config file.

► Run program on monitored host

This is mapped to a System command action in the corresponding situation, which appears as an "action" attribute in the assessment XML output. Note again that consideration should be taken because the user, environment variables, and current working directory may not be the same as the original source Distributed Monitoring monitor.

► Run program on another host

This is not mapped. However, the situation can have this information manually added afterward with the "Execute the action at the managing system (TEMS)" option enabled for the situation.

- Run task

  This is not mapped. However, the task script could be retrieved and used in a "System command" action for the situation.

- Change-icon response

  This is implicitly handled automatically, because the situation would update the status icons in the navigator view automatically, and would also show an update in the situation event console view as well.

- Tivoli Desktop pop-up

  There is not an equivalent GUI-based popup, but a similar notification scheme is available in the TEP Client for playing a sound instead.

- E-mail, notice, and log-to-file responses

  These are not mapped. Alternatives include using the Universal Message action in a situation to update the Universal Message Console.

Assessment and upgrade

When considering assessment and upgrade at this stage, there are two main approaches:

1. Assess individual profiles first, and upgrade the profiles afterward. Then assess profile managers, and upgrade the profile managers afterward.

   Assessing and upgrading profiles first enables monitors to be mapped to situations so that situations are created but are not actually running on any Agents. Within the profile XML assessment output, for each mapped situation, there will be attributes for "distribution" that include placeholders for new managed system list (MSL) names. But because the profile managers are not assessed yet at that point, the MSLs are not created. Assessing and upgrading this way can be helpful for those wanting to separate the profile upgrade phase from actual startup of the situations on the Agents, since it allows the situations to be created without actually running them. Then afterward, the profile managers can be assessed and upgraded to complete the assignment of managed system lists to situations, causing them to be activated on the managed systems.

2. For a smaller environment with more limited numbers of profiles and profile managers, another approach is to assess profiles and profile managers together first, then upgrade particular profiles and profile managers afterward.

   Assessing profiles and profile managers together in one assessment using proper arguments for the **`witmassess`** command enables analysis of particular original profiles in that profile manager and any endpoint subscribers at the bottom of the branches. Upgrade of the profile managers can then be done selectively with particular profiles and endpoints afterward. When the profiles and profile managers are upgraded together, the situations are both created and assigned managed system lists, meaning that any enabled monitors mapped to situations would be set to “Run at startup” and would activate on the managed systems immediately after.

Best practices can be different depending on the number of resources to analyze. For this example, the first approach is used to assess and upgrade profiles first, then profile managers afterward for illustrative purposes.

To assess and upgrade the profiles and profile managers, assessments of only the original profiles will be done, not profile copies. It is assumed in this case that profile copies in lower profile manager levels of the subscription tree are simply exact copies of the originals at the higher levels. Upgrades of the profiles will then be done to create the new situations. Assessment for the profile managers will then be done using a special “flatten” option so that only the profiles in that profile manager are analyzed, and not the mid-level profile copies in the profile managers below. This will be followed by selective upgrades of profiles in each of the profile manager levels

Note: In most environments that used multi-layer profile manager hierarchies with profile copies at each level, when customizations were required, new specific profiles were created rather instead of editing every profile copy along the way. This means that profile copies were still just exact copies of the top-level original profile, as mentioned above. However, if this approach was not used in an environment, and it is known that profile copies were being edited at multiple levels throughout the hierarchy, then the options to selectively choose only original profiles for assessment and upgrade will not be valid, because the profile copies potentially have unique monitoring content. Also, the option to use the “flatten” argument for profile manager assessment would not be suitable. This should be remembered for environments where profile copies were repeatedly edited at different levels.

Using the diagram in Figure 5-11 on page 275 as our continuing example, we continue with the assumption that we are targeting the EMEA hierarchy. So profiles in that branch of the profile manager hierarchy are of primary concern for assessment and upgrade. Looking at the bottom level of that branch, there are profile copies from pm_WW and pm_EMEA profile managers, and also profiles from pm_EMEA_Italia and pm_EMEA_UK. To further the example, two particular endpoints in that hierarchy are being targeted: football and hpdps2.

Profile assessment

First, original profiles will be assessed. In this case, based on the profile manager hierarchy in Figure 5-11 on page 275, and based on the fact that the two targeted endpoints in this example are both UNIX, the following original profiles are of interest:

- pf_EMEA_Italia
- pf_EMEA_unix
- pf_WW_all
- pf_WW_unix

(Profile copies are not being examined here.) So, for example, assessment for the first profile pf_EMEA_Italia can be done in the following way:

```
witmassess -p pf_EMEA_Italia -f baseline.xml
```

Example 5-21 shows an excerpt of this output.

Example 5-21   Sample witmassess profile output

```
..
...
<Profile aggregateStatus="PARTIAL" name="pf_EMEA_Italia#bootcamp4111-region"
source="SentryProfile:1438770036.1.2242#Sentry::All#" target="N/A">
<Monitor aggregateStatus="INCOMPLETE" name="netcoll" source="Monitor:224202"
target="N/A">
            <Threshold source="Threshold:critical" status="INCOMPLETE"
target="Situation:UX_USNetCol_224202_critical1">
               <TargetSituation application="0000000000000000000000G000000
000000000000000" association="Network" autoStart="true"
description="Unix_Sentry:netcoll:LZ_USNetCol_224202_critical"
distribution="pm_EMEA_Italia_1_2222_KLZ"
eventServer="bootcamp.tivlab.austin.ibm.com:0" eventSeverity="CRITICAL"
multipleIntervals="FALSE" name="LZ_USNetCol_224202_critical"
samplingInterval="15:m">
                  <Conditions>*IF *VALUE Linux_Network.Collisions *GT
1000</Conditions>
               </TargetSituation>
               <TargetSituation
application="00f2000000000000000000000000000000000000000" association="Network"
autoStart="true" description="Unix_Sentry:netcoll:UX_USNetCol_224202_critical"
distribution="pm_EMEA_Italia_1_2222_KUX"
eventServer="bootcamp.tivlab.austin.ibm.com:0" eventSeverity="CRITICAL"
multipleIntervals="FALSE" name="UX_USNetCol_224202_critical1"
samplingInterval="15:m">
                  <Conditions>*IF *VALUE Network.Network_Interface_Name *EQ
Aggregate *AND *VALUE Network.Collisions *GT 1000</itmprf:Conditions>
               </TargetSituation>
            </Threshold>
         </Monitor>
...
```

Example 5-21 on page 298 highlights one particular monitor (a netcoll monitor) from an original profile pf_EMEA_Italia. We can see that this is an original profile and not a profile copy because the Profile “name” attribute does not contain any @ characters. In the Monitor element, the monitor name netcoll appears, along with a source value that includes a number derived from the object ID of the SentryProfile: the `2242` represents the object number from the full $TMR.1.2242 object ID, and the extra `02` after it corresponds to the monitor number or record number within the profile. This number is viewable in **`wlsmon <profile>`** output for each monitor. Example 5-22 corresponds to this profile.

Example 5-22 Sample “wlsmon pf_EMEA_Italia” output showing monitor ID

```
# wlsmon pf_EMEA_Italia

Distribution Actions:(none)
224202  Monitor: Network collisions()
  Timing: Every 15 minutes
  start time specification:2005-10-27 05:10PM
...
```

The monitor ID value is next to the monitor name, `Network collisions()`.

This `224202` value uniquely identifies this particular monitor in this particular original profile. Because it uniquely distinguishes it, it will be used later when determining situation names. This monitor had a “critical” response level set up with responses, so a corresponding situation must be created. It looks up the netcoll monitor name in the $DBDIR/AMX/data/mapping/*.xml files to find a corresponding situation base name, and finds UX_USNetCol. At this point, because it maps to a UNIX system, when it creates the situation, it actually creates two: one for UNIX OS Agents, and one for the Linux OS Agent. This is because while Distributed Monitoring Unix_Sentry monitors ran on UNIX and Linux systems, the corresponding situations must be different for the UNIX Agent and Linux Agent.

The naming convention it uses is:

```
<MappedName>_<MonitorID>_<ResponseLevel>
```

In this example, the <MappedName> is UX_USNetCol for UNIX and LZ_USNetCol for Linux. The <MonitorID> is the 224202 value that comes from the “source” attribute mentioned earlier. The <ResponseLevel> in this case is “critical”. So the full name for the two mapped situations becomes:

```
LZ_USNetCol_224202_critical
UX_USNetCol_224202_critical1
```

This naming convention is used because situation names are limited to a maximum of 31 characters, so an efficient way had to be found to try to keep them unique. If the ResponseLevel would make the name longer than 31 characters, the name is truncated at the end to allow it to fit.

If profile copies were being assessed, then the MonitorID portion of the mapped situation name would include additional numbers related to the profile copy object ID; the resulting longer name would distinguish the situation in the profile copy from the situations mapped from the original profile or from other profile copies. In this assessment, profile copies are identical to the original profile, so additional mapped situations would simply be duplicates, and they are not being assessed.

Note that the mapped UNIX situation has a number 1 appended to it. In cases where a single monitor response level ("critical" in the example above) is being mapped to more than one situation (as it is for Linux and UNIX in the example above), there is not a guarantee that the prefix will necessarily be different in all cases for all monitors. If the prefix was not unique, and the same MonitorID and ResponseLevel were appended, there would be a conflict because both names would have been the same. To avoid the possibility of conflicts, a numeric value is appended afterward.

In the case of monitors from the Unix_Sentry collection, the mapped Linux situation is usually created first, and the mapped UNIX situation second; this means that most UNIX monitors mapped to situations will have a 1 value appended to the name. If the name is already 31 characters, the last character will be dropped and replaced with the 1.

In the assessment XML output, for the TargetSituation being reviewed, there is a corresponding "description" attribute; this attribute can be up to 63 characters, so it typically uses the following longer format:

```
<MonitoringCollection>:<monitor>:<name>
```

<MonitoringCollection> is the name of the source Distributed Monitoring Monitoring Collection, <monitor> is the name of the monitor in that collection, and <name> is similar to the mapped situation name but without being truncated or having any characters appended. So, for example, in the sample output above for the netcoll monitor, it was:

```
Unix_Sentry:netcoll:UX_USNetCol_224202_critical
```

As long as it is understood how to derive the situation name using this information, users can review the upgraded situation names and trace them back to the source Distributed Monitoring monitors to be sure where they originated.

Some more tips and special cases regarding these mapped situations:

- Monitors that map successfully to OS Agent Situations usually begin with UX_*, LZ_*, and NT_* characters in their new situation names.

- Monitors that do not map because they are deprecated will appear in the assessment XML output similar to Example 5-23.

Example 5-23 XML output

```
<Monitor aggregateStatus="COMPLETE" comment="AMXUT7508W The specified monitor
was not upgraded because it is deprecated." name="filechk"
source="Monitor:224207" target="N/A"/>
```

- Monitors that do not meet the two previous criteria are considered custom. There are a few important cases to outline here:
  - Custom numeric/string script monitors

    The ncustom and scustom Distributed Monitoring monitors are handled via the Universal Agent. Situations are created using this naming convention:

    ```
    <ncustom|scustom>_<UniqueMonitorID>_<ResponseLevel>
    ```

    For example, in the pf_EMEA_Italia profile assessment XML output, the profile pf_EMEA_Italia has a custom numeric script monitor (Example 5-24).

Example 5-24 Sample custom numeric script assessment output

```
...
<Monitor aggregateStatus="INCOMPLETE" name="ncustom" source="Monitor:224205"
target="N/A">
            <Threshold source="Threshold:critical" status="INCOMPLETE"
target="Situation:ncustom_224205_critical">
               <TargetSituation
application="00000000000008000000000000000000000000000" association="NUMERIC"
autoStart="true" description="Universal:ncustom:ncustom_224205_critical"
distribution="pm_EMEA_Italia_1_2222_KUM"
eventServer="bootcamp.tivlab.austin.ibm.com:0" eventSeverity="FATAL"
multipleIntervals="FALSE" name="ncustom_224205_critical"
samplingInterval="15:m">
<Conditions>*IF *VALUE DM37XNUMERIC00.MonitorKey *EQ 22
4205 *AND *VALUE DM37XNUMERIC00.Stdout *GT 10</Conditions>
               </TargetSituation>
            </Threshold>
            <Threshold source="Threshold:severe" status="INCOMPLETE"
target="Situation:ncustom_224205_severe">
               <TargetSituation
application="00000000000008000000000000000000000000000" association="NUMERIC"
autoStart="true" description="Universal:ncustom:ncustom_224205_severe"
distribution="pm_EMEA_Italia_1_2222_KUM"
eventServer="bootcamp.tivlab.austin.ibm.com:0" eventSeverity="CRITICAL"
multipleIntervals="FALSE" name="ncustom_224205_severe" samplingInterval="15:m">
```

```
                  <Conditions>*IF *VALUE DM37XNUMERIC00.MonitorKey *EQ 224205
*AND *VALUE DM37XNUMERIC00.Stdout *GT 5</Conditions>
               </TargetSituation>
            </Threshold>
...
```

The corresponding mapped situation names here are `ncustom_224205_critical` and `ncustom_224205_severe`, following the naming convention mentioned above.

Note the highlighted entries above for the situation formula, in the Conditions section. It uses a MonitorKey value to distinguish this monitor from other monitors that might be running in the same Universal Agent. Later in the assessment XML output, it shows a TargetScript section.

Example 5-25 Custom numeric script assessment output showing TargetScript

```
...
<TargetScript interval="900" origMonitor="ncustom" origMonitorKey="224205"
origMonitorType="NUMERIC" origProfile="1438770036.1.2242">
               <TargetScriptArgument
argValue="/usr/scripts/tivoli/CheckTivErrors.sh"/>
...
```

The origMonitorKey attribute here corresponds to the monitorKey value used above (224205); this maps the custom script to the right situation to evaluate the results. The upgrade section later in the chapter provides additional details for how this actually works on the Universal Agent.

– Asynchronous monitors

These are also mapped to the Universal Agent. The naming convention is similar to custom scripts:

```
<nasync|sasync>_<UniqueMonitorID>_<ResponseLevel>
```

Example 5-26 shows sample profile assessment output.

Example 5-26 Asynchronous monitor profile assessment output example

```
<Monitor aggregateStatus="INCOMPLETE" name="nasync" source="Monitor:224206"
target="N/A">
      <Threshold source="Threshold:critical" status="INCOMPLETE"
target="Situation:nasync_224206_critical">
         <TargetSituation
application="00000000000008000000000000000000000000000"
association="NUMERICASYNC" autoStart="true" description="Universal:nasync:na
sync_224206_critical" distribution="pm_EMEA_Italia_1_2222_KUM"
eventServer="bootcamp.tivlab.austin.ibm.com:0" eventSeverity="FATAL"
multipleIntervals="FALSE" name="nasync_224206_critical"
samplingInterval="15:m">
```

```
            <Conditions>*IF *VALUE DM37XNUMERICASYNC00.MonitorKey *EQ 224206
*AND *VALUE DM37XNUMERICASYNC00.Stdout *EQ 911</Conditions>
         </TargetSituation>
      </Threshold>
      <Threshold source="Threshold:warning" status="INCOMPLETE"
target="Situation:nasync_224206_warning">
         <TargetSituation
application="00000000000008000000000000000000000000000"
association="NUMERICASYNC" autoStart="true"
description="Universal:nasync:nasync_224206_warning"
distribution="pm_EMEA_Italia_1_2222_KUM"
eventServer="bootcamp.tivlab.austin.ibm.com:0" eventSeverity="MINOR"
multipleIntervals="FALSE" name="nasync_224206_warning" samplingInterval="15:m">
            <Conditions>*IF *VALUE DM37XNUMERICASYNC00.MonitorKey *EQ 224206
*AND *VALUE DM37XNUMERICASYNC00.Stdout *EQ 411</Conditions>
         </TargetSituation>
      </Threshold>
      <TargetScript origMonitor="nasync" origMonitorKey="224206"
origMonitorType="NUMERIC" origProfile="1438770036.1.2242">
         <TargetScriptArgument argValue="chitchat"/>
      </TargetScript>
   </Monitor>
```

The situation names for this numeric asynchronous monitor become `nasync_224206_critical` and `nasync_224206_warning`, following the naming convention mentioned earlier. The argValue `chitchat` corresponds to the asynchronous channel name from Distributed Monitoring. The upgrade section later in the chapter explains how this is handled in the Universal Agent.

– Custom monitors from custom collections

These are also handled by the Universal Agent. Example 5-27 shows sample output for a custom monitor named `custommon`.

Example 5-27 Profile assessment showing custom monitor from custom collection

```
...
   <Monitor aggregateStatus="INCOMPLETE" name="custommon"
source="Monitor:224208" target="N/A">
      <Threshold source="Threshold:critical" status="INCOMPLETE"
target="Situation:custommon_224208_critical">
         <TargetSituation
application="00000000000008000000000000000000000000000" association="STRING"
autoStart="true" description="customcol:custommon:custommon_224208_critical"
distribution="pm_EMEA_Italia_1_2222_KUM"
eventServer="bootcamp.tivlab.austin.ibm.com:0" eventSeverity="MINOR"
multipleIntervals="FALSE" name="custommon_224208_critical"
samplingInterval="1:H">
```

```
            <Conditions>*IF *VALUE DM37XSTRING00.MonitorKey *EQ 224208 *AND
*VALUE DM37XSTRING00.Stdout *EQ "15"</Conditions>
         </TargetSituation>
      </Threshold>
      <Threshold source="Threshold:severe" status="INCOMPLETE"
target="Situation:custommon_224208_severe">
         <TargetSituation
application="00000000000000800000000000000000000000000000" association="STRING"
autoStart="true" description="customcol:custommon:custommon_224208_severe"
distribution="pm_EMEA_Italia_1_2222_KUM" multipleIntervals="FALSE"
name="custommon_224208_severe" samplingInterval="1:H">
            <Conditions>*IF *VALUE DM37XSTRING00.MonitorKey *EQ 224208 *AND
*VALUE DM37XSTRING00.Stdout *NE "15"</Conditions>
         </TargetSituation>
      </Threshold>
      <TargetScript interval="3600" origMonitor="custommon"
origMonitorKey="224208" origMonitorType="STRING"
origProfile="1438770036.1.2242">
         <TargetScriptProgram flagsOrArguments="-c" interpreter="/bin/sh"
interps="aix4-r1   w32-ix86   hpux10   solaris2"
programPath="#!/bin/sh%CR%sleep 55%CR%echo \&quot;ok\&quot;%CR%exit
0%CR%%CR%"/>
      </TargetScript>
   </Monitor>
...
```

Note that for a situation mapped from a custom monitor like this, the naming convention used is:

```
<MonitorName>_<MonitorID>_<ResponseLevel>
```

Note also that unlike the ncustom, scustom, nasync, and sasync monitors that were mapped earlier, this custom monitor must have the monitoring implementation embedded in the XML. This is because in most cases, custom monitors from custom collections typically have the scripts embedded in them. The script is highlighted above in the TargetScriptProgram element.

Profile upgrade

Each of the profiles can then be upgraded using **witmupgrade** commands; for example, with a current working directory of $DBDIR/shared/analyze:

```
   witmupgrade -x profiles/pf_EMEA_Italia.xml
   witmupgrade -x profiles/pf_EMEA_unix.xml
   witmupgrade -x profiles/pf_WW_all.xml
   witmupgrade -x profiles/pf_WW_unix.xml
```

Following the upgrade, any new situations that were created for monitors and response levels are updated in the $DBDIR/AMX/data/status.properties file. This is the same file that is updated by successful endpoint upgrades. Full upgrade results are stored in XML format in the $DBDIR/AMX/shared/analyze/profiles/upgrade directory, in a file with the naming convention <profile>_<timestamp>.xml to distinguish it from previous upgrade attempts. The output can be examined to determine whether the profile overall has been upgraded, and if so, what situations were created. Example 5-28 shows sample output for the pf_EMEA_Italia profile.

Example 5-28 Profile upgrade XML output

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Copyright IBM Corporation 2005-->
<Profile aggregateStatus="COMPLETE" name="pf_EMEA_Italia"
source="SentryProfile:1438770036.1.2242#Sentry::All#" target="N/A"
xmlns="http://www.ibm.com/tivoli/itm/prf"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.ibm.com/tivoli/itm/prf profile.xsd">
   <Monitor aggregateStatus="COMPLETE" name="netcoll" source="Monitor:224202"
target="N/A">
      <Threshold source="Threshold:critical" status="COMPLETE"
target="Situation:UX_USNetCol_224202_critical1">
         <TargetSituation
application="000000000000000000000G00000000000000000000" association="Network"
autoStart="true" description="Unix_Sentry:netcoll:LZ_USNetCol_224202_critical"
distribution="pm_EMEA_Italia_1_2222_KLZ"
eventServer="bootcamp.tivlab.austin.ibm.com:0" eventSeverity="CRITICAL"
multipleIntervals="FALSE" name="LZ_USNetCol_224202_critical"
samplingInterval="15:m">
            <Conditions>*IF *VALUE Linux_Network.Collisions *GT
1000</Conditions>
         </TargetSituation>
...
```

A `status="COMPLETE"` attribute in the Threshold element indicates that a mapped situation was created successfully. An aggregateStatus="COMPLETE" attribute in the Monitor element indicates that all response levels for that particular monitor were mapped to situations that were created successfully; a PARTIAL value indicates that only some had situations created successfully, and INCOMPLETE indicates that none of the situations had been created. In the latter two cases, logs and traces on the TMR server for the profile upgrade should be examined relative to those situation names to try to determine what had failed ($DBDIR/AMX/logs/log_witmupgrade_<timestamp>.xml, where <timestamp> matches the timestamp used in the upgrade XML output filename).

Note that for monitors that were mapped to situations for the Universal Agent, the profile upgrade leads to a number of files being updated on the target managed system where the Universal Agent is running. For example, the Universal Agent metafile in $CANDLEHOME/<platform>/um/metafiles/dm37x/dm37x.mdl is updated for each of the ncustom, scustom, nasync, and sasync mapped situations. Example 5-29 shows one section of this file.

Example 5-29 Sample dm37x.mdl

```
...
...
//Name String K 3600 AddTimeStamp @String attribute group used for String
script
 monitors upgraded from Distributed Monitoring.
//Source script /bin/sh /usr/IBM/ITM/aix516/um/work/dm37x/script/224208 ""
envfile=/usr/IBM/ITM/aix516/um/work/dm37x/sync/224208.env    Interval=3600
ManagedSystemName=football_224208
//Attributes
MonitorKey      (GetEnvValue = MONITOR_ID) @Monitor key derived from
Distributed Monitoring.
MonitorName     (GetEnvValue = PROBE) @Monitor name derived from Distributed
Monitoring.
ProfileName     (GetEnvValue = PROFILE_NAME)@Profile Name derived from
Distributed Monitoring.
Stdout  Z  512 -FILTER={MATCH(0,#)} @String value returned by custom script.
...
...
```

The `Source script` entry shows the script that is called locally from the $CANDLEHOME/<platform>/um/work/dm37x/script directory. The envFile attribute is built using information taken from the $DBDIR/AMX/data/environments files on the TMR server for whichever profile was involved. The `Interval` shows 3600, meaning that the data will be presentable within the TEP client for up to 3600 seconds (1 hour). Finally, the `ManagedSystemName` can be used to determine what navigator item this will appear under; in this case the name `football_224208` means it will appear as the football_224208:DM37X00 entry in the navigator.

These files were pushed via an EPA_deployScript downcall sent to the endpoint, while the remaining activity of creating situations is handled via SOAP calls to the Hub TEMS. So tracing of problems involves reviewing the lcfd.log on the endpoint, or Hub TEMS trace files, beyond those files already mentioned in $DBDIR/AMX/logs, $DBDIR/AMX/trace, and the temporary wtemp directory on the TMR server.

Profile manager assessment and upgrade

At this stage, the assessment and upgrade of profiles has led to the creation of situations in the V6.1 environment. However, these situations do not have managed system lists associated that have actual managed systems in them yet. The profile manager assessment and upgrade is the final step that generates the proper managed system list content.

Before assessing and upgrading, it is important to understand the overall implications of profile manager assessment. When the **witmassess** command is used to assess profile managers using the -pm <ProfileManagerName> option, it also automatically assesses the profiles that are within it. By default, when the -o flatten option is not used, the assessment recursively examines the indicated profile manager and all nested profile managers. When it does this, it also automatically examines all original profiles and all profile copies in each profile manager. So, for example, consider a basic command:

```
witmassess -pm pm_WW -f baseline.xml
```

It would assess the following profiles and profile managers for all nested profiles and profile managers in the pm_US tree, including:

pm_WW	pf_WW_all, pf_WW_unix, pf_WW_windows
pm_EMEA	pf_EMEA_unix, pf_WW_unix@pm_EMEA, pf_WW_windows@pm_EMEA, pf_WW_all@pm_EMEA
pm_EMEA_Italia	pf_EMEA_Italia, and profile copies from EMEA...
pm_EMEA_UK	pf_EMEA_UK, and profile copies from pm_EMEA...

All of this content would be embedded in one file, $DBDIR/AMX/shared/analyze/profilemanagers/pm_WW.xml. In one sense this is useful, but after actually perusing the content to determine what has been identified, even with customized XSL stylesheets or other formatting tools, this can become difficult, especially when the resources are being updated in phases. Additionally, many users do not edit the profile copies that are generated in nested profile managers. Instead, new profiles are created in those levels. For the example above, the assessment would include all profiles and profile copies, all in one very large XML file.

To avoid these two issues, instead of using the basic format above, the -o flatten option is added where useful, and the assessment commands are separated into three logical steps. Also, since only the original profiles are of interest, the command can include arguments for those original profiles that were assessed and upgraded earlier:

```
witmassess -pm pm_EMEA_Italia -p pf_EMEA_Italia#bootcamp4111-region -f
scans/baseline.xml
witmassess -pm pm_EMEA -p pf_EMEA_unix#bootcamp4111-region -f
scans/baseline.xml -o flatten
witmassess -pm pm_WW -f scans/baseline.xml -o flatten
```

Note: For the version of code used when the book was being written, combining the -pm argument and -p argument in the **witmassess** command only worked if the fully qualified profile name was used with the -p argument; thus, the region name bootcamp4111-region is appended along with a # separator character when specifying the profile names above. If this was not done, the assessment completed without errors but the assessment XML output did not actually assess the profile properly. If attempts to specify both -pm and -p result in assessment XML output that has empty content for the Profiles element, try using the full profile name including the region name as done above in the **witmassess** command.

For the profile manager pm_EMEA_Italia, the -o flatten option is not needed as it is at the bottom level already. Example 5-30 shows sample output.

Example 5-30 Sample profile manager assessment output

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Copyright IBM Corporation 2005-->
<ProfileManager aggregateStatus="PARTIAL"
source="ProfileManager:pm_EMEA_Italia"
 target="N/A" xmlns="http://www.ibm.com/tivoli/itm/pm"
xmlns:ep="http://www.ibm.com/tivoli/itm/ep"
xmlns:itmprf="http://www.ibm.com/tivoli/itm/prf"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.ibm.com/tivoli/itm/pm pm.xsd">
   <Profiles aggregateStatus="PARTIAL" source="Profiles" target="N/A">
      <itmprf:Profile aggregateStatus="PARTIAL"
name="pf_EMEA_Italia#bootcamp4111-region"
source="SentryProfile:1438770036.1.2242#Sentry::All#" target="N/A">
...
...
```

Upgrade of the profile manager can then be done using the assessment XML as input; for example, from the current working directory of $DBDIR/AMX/shared/analyze:

```
witmupgrade -x profilemanagers/pm_EMEA_Italia.xml -u -f scans/baseline.xml
witmupgrade -x profilemanagers/pm_EMEA.xml -u -f scans/baseline.xml
witmupgrade -x profilemanagers/pm_WW.xml -u -f scans/baseline.xml
```

The upgrade of the profile manager leads to the generation of the managed system lists mentioned in the assessment XML, and the appropriate managed systems are added. Note that one of the reasons the IBM JRE 1.4.2 is needed for the Distributed Monitoring Upgrade tools is that it contains needed functionality for allowing these MSLs to be updated properly. So, if it was not being used, one of the limitations would be the inability to automatically create and populate the MSLs.

Example 5-31 shows sample upgrade output.

Example 5-31 Sample profile manager upgrade output

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Copyright IBM Corporation 2005-->
<ProfileManager aggregateStatus="COMPLETE" source="ProfileManager:pm_EMEA"
target="N/A" xmlns="http://www.ibm.com/tivoli/itm/pm"
xmlns:ep="http://www.ibm.com/tivoli/itm/ep"
xmlns:itmprf="http://www.ibm.com/tivoli/itm/prf"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.ibm.com/tivoli/itm/pm pm.xsd">
   <Profiles aggregateStatus="COMPLETE" source="Profiles" target="N/A">
      <itmprf:Profile aggregateStatus="COMPLETE"
name="pf_EMEA_unix#bootcamp4111-region"
source="SentryProfile:1438770036.1.2241#Sentry::All#" target="N/A">
         <itmprf:Monitor aggregateStatus="COMPLETE" name="netout"
source="Monitor:224100" target="N/A">
            <itmprf:Threshold source="Threshold:critical" status="COMPLETE"
target="Situation:UX_USNetOut_224100_critical1">
               <itmprf:TargetSituation
application="000000000000000000000G000000000000000000" association="Network"
autoStart="true" description="Unix_Sentry:netout:LZ_USNetOut_224100_critical"
distribution="pm_EMEA_1_2224_KLZ"
eventServer="bootcamp.tivlab.austin.ibm.com:0" eventSeverity="CRITICAL"
multipleIntervals="FALSE" name="LZ_USNetOut_224100_critical"
samplingInterval="1:H">
                  <itmprf:Conditions>*IF *CHANGE
Linux_Network.Packets_Transmitted *GE 100000</itmprf:Conditions>
               </itmprf:TargetSituation>
```

The aggregateStatus=”COMPLETE” for the ProfileManager element will be set only if all assessed profiles and subscribers under it have aggregateStatus=”COMPLETE” as well. At this stage, because all desired endpoints and profiles have been assessed and upgraded, the profile manager assessment is essentially just getting the MSLs set up.

The naming convention for MSLs is:

```
<ProfileManagerName>_<PartialObjectID>_<AgentType>
```

<ProfileManagerName> is the text name of the profile manager.
<PartialObjectID> is the last two numbers of the object ID for the profile manager (confirmable using **`wlookup -or ProfileManager <ProfileManagerName>`**), with the periods converted to underscore characters so that they are valid.

<AgentType> is either KLZ (Linux), KUX (UNIX), KNT (Windows), or KUM (Universal Agent). For example, for the profile manager pm_EMEA above, the object ID was $TMR.1.2224; this results in a PartialObjectID value of 1_2224. The final name for the mapped MSLs becomes pm_EMEA_1_2224_KLZ for any Linux OS Agent members and pm_EMEA_1_2224_KUX for UNIX OS Agent members.

Reassess profiles and profile managers

Reassessment can be done to confirm that the status was properly updated after completion of upgrading profiles and profile managers. For example, for the profile manager pm_EMEA_Italia and profile pf_EMEA_Italia:

```
witmassess -pm pm_EMEA_Italia -p pf_EMEA_Italia#bootcamp4111-region -f
scans/baseline.xml
```

The XML assessment output in Example 5-32 is generated, showing it to be complete.

Example 5-32 Reassessment of profile manager, showing COMPLETE status

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Copyright IBM Corporation 2005-->
<ProfileManager aggregateStatus="COMPLETE"
source="ProfileManager:pm_EMEA_Italia
" target="N/A" xmlns="http://www.ibm.com/tivoli/itm/pm"
xmlns:ep="http://www.ibm.com/tivoli/itm/ep"
xmlns:itmprf="http://www.ibm.com/tivoli/itm/prf"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.ibm.com/tivoli/itm/pm pm.xsd">
   <Profiles aggregateStatus="COMPLETE" source="Profiles" target="N/A">
      <itmprf:Profile aggregateStatus="COMPLETE"
name="pf_EMEA_Italia#bootcamp4111-region"
source="SentryProfile:1438770036.1.2242#Sentry::All#" target="N/A">
         <itmprf:Monitor aggregateStatus="COMPLETE" name="netcoll"
source="Monitor:224202" target="N/A">
            <itmprf:Threshold source="Threshold:critical" status="COMPLETE"
target="Situation:UX_USNetCol_224202_critical1">
               <itmprf:TargetSituation
application="0000000000000000000000G00000
000000000000000" association="Network" autoStart="true"
description="Unix_Sentry:netcoll:LZ_USNetCol_224202_critical"
distribution="pm_EMEA_Italia_1_2222_KLZ"
eventServer="bootcamp.tivlab.austin.ibm.com:0" eventSeverity="CRITICAL"
multipleIntervals="FALSE" name="LZ_USNetCol_224202_critical"
samplingInterval="15:m">
...
```

Alternatives

The previous example detailed one branch of a multi-level profile manager hierarchy. There are other command flows that could be used, depending on environment configuration.

Smaller single-TMR environments

One feasible alternative for smaller environments is:

1. Assess and upgrade endpoints first as usual.
2. Instead of assessing and upgrading profiles separately, and assessing and upgrading profile managers separately, use **`witmassess`** to do the assessments of profile managers and profiles together, and use that assessment XML as input to the **`witmupgrade`**.

If this is done, the creation of profiles and managed system lists will occur in the same pass of **`witmupgrade`**, rather than keeping the two major functions uncoupled. Essentially, phased manageability of the upgrade is given up in the interest of speed.

Hub-Spoke interconnected TMRs

In Hub-Spoke interconnected TMR environments, profiles are centrally managed, created, and edited in the Hub TMR, and distributed to profile managers that have endpoints subscribed from the spoke TMRs. The Hub centrally manages the profiles, and the spokes separately manage the resources being monitored. The overall flow of assessment and upgrade is the same, but a few steps in-between differ:

- Installation

  The Upgrade Toolkit still has to be installed on the TMR server and managed nodes in both the Hub and the Spoke TMRs.

- Scan TMR

  All TMRs must still be scanned using **`witmscantmr`**.

- Install infrastructure

  Using the baseline XML output from each TMR, an infrastructure should be installed to handle the load. Afterward, another scan of the TMRs can be done to confirm the updated status in the baseline files. The baseline XML files can then be copied from the Spoke TMRs to the Hub TMR in the $DBDIR/AMX/shared/analyze/scans directory for use later. Leave the default file names intact so that they can be identified later during profile and profile manager assessment.

- Assess endpoints

  The endpoints are typically all in the Spoke TMRs, so the endpoint assessment should be done only in the Spoke TMRs.

- Upgrade endpoints

  Upgrade endpoints in the Spoke TMRs.

  Recall that this generates a number of environment files used later by custom monitors, in the $DBDIR/AMX/data/environment directory. These should be copied to the same directory on the Hub TMR server because they will be needed for the profile upgrade step. They will be needed on the Hub TMR server because the profile assessment and upgrade will only be done from the Hub TMR; it will only be done at the Hub TMR server because the original profiles are in the Hub TMR, and only profile copies exist in the spoke TMRs.

- Assess profiles

  This is done from the Hub TMR server only, since original profiles only exist in the Hub TMR; this helps avoid additional unnecessary duplicate situations from being generated from the profile copies. Assessment should be done without using the -f <baselineXML> option; this enables multiple baseline XMLs to be identified based on region number.

- Upgrade profiles

  This is done only from the Hub TMR server, as assessment only ran there.

- Assess profile managers

  This only occurs from the Hub TMR server because it does not require traversing the profile manager hierarchy in the Spoke TMRs, which would lead to unnecessary MSLs with lists of targets that duplicated other MSLs. The -o flatten argument can be used so that only the top profile manager and its profiles are assessed. Remember that the -p profile#region argument can also be added to further target particular profiles only. Also, remember that although the -o flatten option eliminates profile manager and profile assessment in middle-level profile managers, the full subscriber hierarchy is still traversed so that the proper endpoint subscriber list can be ascertained. Finally, assessment should be done without using the -f <baselineXML> option; this enables multiple baseline XMLs to be identified based on region number.

- Upgrade profile managers

  This is done only from the Hub TMR server because the assessment was only done there. If the assessment was further limited to particular profiles using -p profile#region in the profile manager assessment, then the upgrade will use that assessment XML as input, and will subsequently only upgrade those particular profiles.

5.4 Post-upgrade considerations

Optional commands are available to roll back upgraded resources in the IBM Tivoli Monitoring 6.1 environment, or to disable monitors from the source Distributed Monitoring environment. This section discusses these, along with base differences in command line interfaces (CLIs) between operations in Distributed Monitoring and IBM Tivoli Monitoring 6.1.

5.4.1 Cleanup

After upgrading resources into the IBM Tivoli Monitoring 6.1 environment, you may find that you want to roll back changes that were made in the new 6.1 environment. Or, if changes are considered acceptable, a user may want to disable the old Distributed Monitoring monitors that were upgraded successfully so that duplicate monitoring is not occurring in the old environment. This section details how both can be done, as well as additional cleanup afterward.

Rollback 6.1 changes

Following an upgrade using the **`witmupgrade`** command, changes that have been made can be rolled back using the **`witmupgrade`** -r option:

- Identify the resources to be rolled back. The most common example is if the changes that were made involved new situations that were created that were not wanted when a profile was upgraded; in this case, identify the source profile assessment XML that was used in the $DBDIR/AMX/shared/analyze/profiles directory.
- Use **`witmupgrade -x <AssessXMLFile> -r -f <baselineXML>`** to roll back the changes. In the example mentioned above for rollback of the upgraded profile resources, the following general process occurs:
  - Updates that were made to tecserver.txt for upgraded situations are removed from the tecserver.txt file on the Hub TEMS. This is done via EPA_setEventForwarding downcalls to the endpoint running on the Hub TEMS.
  - Upgraded situations mentioned in the assessment XML output are removed via SOAP calls to the Hub TEMS.
  - Upgraded custom situation files such as scripts and environment files are removed from the Universal Agent via EPA_removeScript downcalls through the endpoint.

Disable Distributed Monitoring monitors

The most common case for this is if monitors were successfully upgraded from profiles to situations and you want to disable those monitors in the old Distributed Monitoring engines. The **witmupgrade** command can be used to do this:

- Identify the endpoint that needs to have Distributed Monitoring monitors disabled in the engine.
- Use **witmupgrade -x <AsessXMLFile> -c -f <baselineXML>**, where <AssessXMLFile> is the endpoint assessment XML file name containing those monitors that need to be disabled. In this case, an mpe_disable downcall is done to the endpoint to disable all monitors on the endpoint. This is not selective: Programatically, it does the equivalent of **wdisprb -z "*" "*" "*" <endpoint>**.

> **Note:** This disables the monitors, and they stay disabled as long as the engine is not stopped and restarted. However, if the engine is restarted, the old enabled/disabled setting is reloaded from persistent store, so the monitors would start running again.
>
> If the engine and all monitors must be uninstalled completely, this can be done manually. For an example of manually disabling and uninstalling Distributed Monitoring from a particular endpoint, see:
>
> http://www.ibm.com/support/docview.wss?uid=swg21048249

The **witmupgrade** can also be used to selectively disable monitors that originated from particular profiles. This would be done using profile assessment XML files with the -x argument to **witmupgrade**. However, when doing this, the monitors are only disabled in the profiles themselves, not in the engines where those profiles had been distributed. If only certain profiles have to be disabled in the engines, then the profiles have to be redistributed to the target endpoints after the **witmupgrade -x <ProfileAssessmentXML> -c -f baseline.xml** was done.

Additional considerations

Additional considerations should be made throughout the upgrade process that potentially affect deployment, including:

- OS Agents and predefined situations set to “Run at startup”: Following endpoint upgrade, even before any profiles have been upgraded, OS Agents typically have a number of predefined situations, some of which are set to Run at startup. This behavior differs slightly from what a seasoned TME-based and Distributed Monitoring-based Tivoli administrator might expect. Such administrators typically are used to default monitors and profiles, but they are not activated unless they have been distributed. Having predefined monitoring settings that are activated on Agent startup may be a surprise to administrators in these cases. The Agent User's Guides or the TEP client Situation Editor should be reviewed to confirm which ones are predefined and set to run this way to understand what will be monitored immediately following OS Agent deployment.

- Throughout the discussion, it has been assumed that the profile manager hierarchy and profile content has been in a relatively steady-state condition, such that XML output would not significantly change over the course of assessment and upgrade. The tools do allow the ability to reassess and re-upgrade selectively over time to deal with these changes. However, substantial changes to the Distributed Monitoring environment during assessment and upgrade causes difficulties in interpretation of XML content and potential unanticipated errors. If extensive changes are planned for the old Distributed Monitoring environment while assessment and upgrade are implemented, consideration should be made as to whether those changes could be better handled directly in the new IBM Tivoli Monitoring 6.1 environment. This helps avoid the complications from volatile assessment and upgrade output, and facilitates efficient monitoring of resources by avoiding the need to implement monitoring in the old Distributed Monitoring environment and later having to upgrade it to the new 6.1 environment in a second step.
- The OS versions supported by Distributed Monitoring may differ from the final list of supported versions for IBM Tivoli Monitoring 6.1. The hardware, OS, and software requirements should be reviewed first to ensure feasibility and supportability of the upgraded Agents in the new 6.1 environment.
- A common heartbeat monitoring implementation for Distributed Monitoring involves running monitors that simply do an upcall to run a program on a central managed node such as the TMR server, where the program touches a particular file that corresponds to the endpoint. The TMR server then has an endpoint Distributed Monitoring engine running a monitor that reviews the timestamps of these files to determine which ones are old, and thus which endpoint engines might be having problems. When we wrote this book, because of the differences in implementations for this heartbeat it was not clear whether all implementations would be upgraded seamlessly to the new OS Agent / Universal Agent structure. As heartbeating already occurs between TEMS and their connected Agents, alternatives might be considered for tracking Agent status differently in 6.1 to avoid problems migrating this older Distributed Monitoring heartbeat implementation.

Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint

This chapter provides a description of the features and components of the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint.

The Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint extends the capability of IBM Tivoli Monitoring 5.1.2 Fix Pack 6, enabling data collected by deployed IBM Tivoli Monitoring 5.1.2 Fix Pack 6 endpoints to be displayed in the IBM Tivoli Monitoring 6.1 Tivoli Enterprise Portal and stored in the IBM Tivoli Monitoring 6.1 Tivoli Data Warehouse component.

6.1 Introduction

The new IBM Tivoli Monitoring 6.1 introduced a complete new architecture for monitoring based on the old OMEGAMON. The connection to existing environments is done with different levels of integration to the main affected products. These integration interfaces are part of IBM Tivoli Monitoring 6.1:

- Tivoli Enterprise Console
- IBM Tivoli Monitoring 5.x
- OMEGAMON

They can be combined as described in Figure 6-1. In this chapter we discuss IBM Tivoli Monitoring 5.x integration.

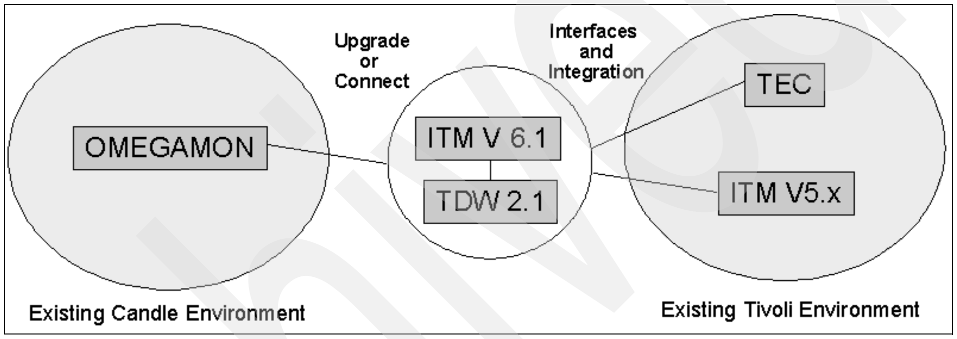

Upgrade or Connect
Interfaces and Integration
OMEGAMON
ITM V 6.1
TDW 2.1
TEC
ITM V5.x
Existing Candle Environment
Existing Tivoli Environment

Figure 6-1 Integration interfaces part of IBM Tivoli Monitoring 6.1

6.2 Features of the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint

IBM Tivoli Monitoring 6.1 is the base software for the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint.

IBM Tivoli Monitoring provides a way to monitor the availability and performance of all systems in your enterprise from one or several designated workstations. It also provides useful historical data that you can use to track trends and to troubleshoot system problems.

The Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint provides the following functions:

- Easy to use, highly customizable Tivoli Enterprise Portal: The Tivoli Enterprise Portal displays information through the use of workspaces in the form of charts and tables. You can start monitoring activity and system status immediately with the predefined workspaces. With just a few clicks of the mouse, you can tailor workspaces to look at specific conditions, display critical threshold values in red, and filter incoming data so you see only what matters. You can also change the hierarchical order in which agents are displayed, enabling you to customize the hierarchy as appropriate for your business. See *IBM Tivoli Monitoring, Version 6.1.0 User's Guide*, SC32-9409, for information. When using Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint, the portal displays the data for the IBM Tivoli Monitoring 5.1.2 resource models including their health, indication health and values from the indications, and status. This agent collects and displays the real-time raw (historical) data logged by the IBM Tivoli Monitoring 5.1.2 resource models.

- IBM Tivoli Monitoring 5.x Data Visualization: Using the IBM Tivoli Monitoring 5.x Endpoint agent, you can view data from IBM Tivoli Monitoring 5.x resource models in the Tivoli Enterprise Portal, as well as warehouse granular data in the Tivoli Data Warehouse. You can use this visualization to replace the Web Health Console used in IBM Tivoli Monitoring 5.1. For information about using this visualization, see *IBM Tivoli Monitoring: IBM Tivoli Monitoring 5.x Endpoint Agent User's Guide*, SC32-9490.
- Integration of multiple monitoring sources in one view: Using the Tivoli Enterprise Portal, you can view monitoring data for your z/OS resources, as well as the applications monitored by the IBM Tivoli Composite Application Manager for Response Time Tracking product, in addition to the data collected by the IBM Tivoli Monitoring agents.
- Support for a Linux Tivoli Enterprise Portal Server and Tivoli Enterprise Portal desktop client: Both Intel Linux and Linux for z/OS are supported operating systems for the Tivoli Enterprise Portal Server and desktop client. (This is a change from OMEGAMON 350.)
- Easy to use Tivoli Data Warehouse with granular historical data rollup: The new Tivoli Data Warehouse enables you to enable and disable history collection for different components and attribute groups, define the periodic time period to save data, define how you want to summarize your historical data, and define how and when you want to prune your historical and detailed data, all through the Tivoli Enterprise Portal. For information about the Tivoli Data Warehouse, see the *IBM Tivoli Monitoring Installation and Setup Guide*, GC32-9407. The Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint stores the raw data logged by the IBM Tivoli Monitoring 5.1.2 resource models in the IBM Tivoli Monitoring 6.1 Tivoli Data Warehouse component.
- Integrated Warehouse Reporting: You can view historical data, using the same workspaces as real-time monitoring, through the Tivoli Enterprise Portal instead of having to generate a report through a database reporting program, such as Crystal Reports.

- Support for IBM DB2, Oracle, and Microsoft SQL warehouse databases: Support for additional databases removes the inconvenience of needing one brand of database for your monitoring environment and a different brand for other applications. The Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint displays the historical data stored in the IBM Tivoli Monitoring 6.1 Tivoli Data Warehouse component in the same views used to display the real-time data.
- IBM Tivoli Enterprise Console integration: You can view IBM Tivoli Enterprise Console events through the Tivoli Enterprise Portal (using the fully functional IBM Tivoli Enterprise Console event console), as well as forward events from IBM Tivoli Monitoring situations to IBM Tivoli Enterprise Console. Event synchronization enables you to send and receive updates made on the event server to the situation events. See Chapter 10, "Integration with IBM Tivoli Enterprise Console" on page 583 for more information about integrating with IBM Tivoli Enterprise Console. The Tivoli Enterprise Portal includes a Situation Event Viewer that displays IBM Tivoli Monitoring events triggered by the situations you are monitoring. If you do not need the enterprise event management capabilities provided by IBM Tivoli Enterprise Console, you can use this integrated event viewer to manage your events.
- Hot Standby for the Tivoli Enterprise Monitoring Server: The Hot Standby feature enables you to maintain failover support for your hub monitoring server by defining a standby monitoring server to act as backup. See *IBM Tivoli Monitoring Installation and Setup Guide*, GC32-9407.
- Administration through the command line: A new set of commands (tacmd option) has been added so you can administer your environment from the command line (both Windows and UNIX). *IBM Tivoli Monitoring Installation and Setup Guide*, GC32-9407, has more about these commands.
- SOAP and SQL interfaces to data: You can create SOAP and SQL scripts to perform actions on the data in your environment. You can deploy the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint to any of your IBM Tivoli Monitoring 5.1.2 Fix Pack 6 endpoints. This function is added to the IBM Tivoli Monitoring 5.1.2 Endpoint, which continues to run all of the deployed resource models using your customized settings.
- Logging data and forwarding it to Tivoli Data Warehouse 1.2 is still supported: After you have deployed the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint, you can choose to send the IBM Tivoli Monitoring engine data to one or both of the following places:
  - To the endpoint database for use by the Web Health Console and for forwarding to Tivoli Data Warehouse 1.2
  - To the IBM Tivoli Monitoring 6.1 system, where it can be viewed in the Tivoli Enterprise Portal and forwarded to the IBM Tivoli Monitoring Tivoli Data Warehouse component

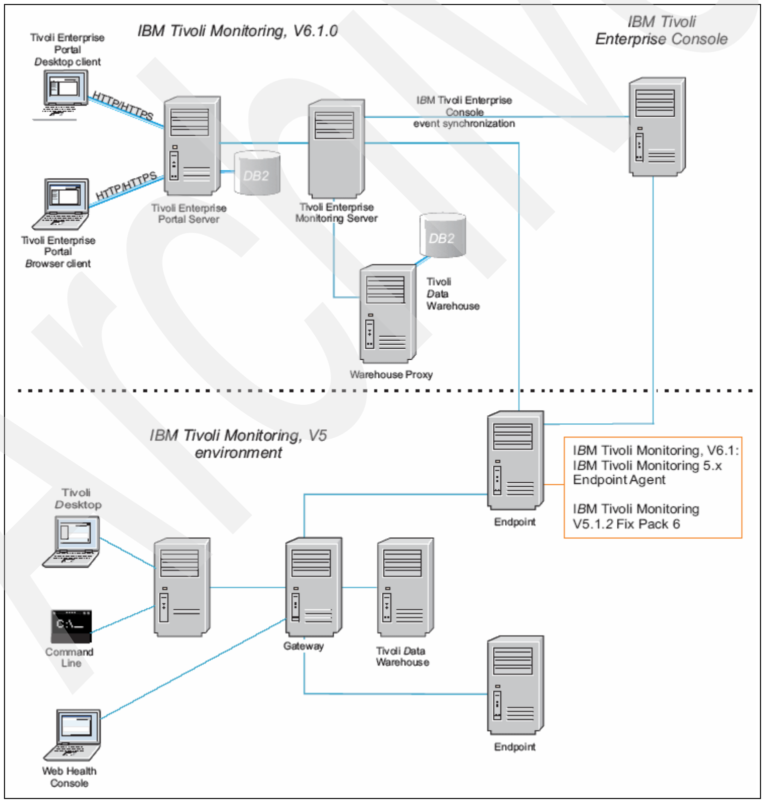
Using the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint enables an installed and running IBM Tivoli Monitoring 5.1.2 environment to send data to Tivoli Enterprise Portal and store data in the IBM Tivoli Monitoring Tivoli Data Warehouse component without disruption.

6.3 Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint components

When using the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint, you are working with two integrated environments:

- ► IBM Tivoli Monitoring 5.1.2 Fix Pack 6
- ► IBM Tivoli Monitoring 6.1

Figure 6-2 shows the components of an IBM Tivoli Monitoring environment that uses the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint.



IBM Tivoli Monitoring, V6.1.0
IBM Tivoli Enterprise Console
Tivoli Enterprise Portal Desktop client
HTTP/HTTPS
HTTP/HTTPS
Tivoli Enterprise Portal Browser client
DB2
Tivoli Enterprise Portal Server
Tivoli Enterprise Monitoring Server
IBM Tivoli Enterprise Console event synchronization
DB2
Tivoli Data Warehouse
Warehouse Proxy
IBM Tivoli Monitoring, V5 environment
Endpoint
IBM Tivoli Monitoring, V6.1: IBM Tivoli Monitoring 5.x Endpoint Agent
IBM Tivoli Monitoring V5.1.2 Fix Pack 6
Tivoli Desktop
Command Line
Gateway
Tivoli Data Warehouse
Endpoint
Web Health Console

Figure 6-2 IBM Tivoli Monitoring V5.x and V6.1 coexistence

The top half of the diagram shows the base components of an IBM Tivoli Monitoring 6.1 environment:

- Tivoli Enterprise Portal client (TEP)

  Tivoli Enterprise Portal is a single point of control for managing the resources your applications rely on, including a range of operating systems, servers, databases, platforms, and Web components.

  The Tivoli Enterprise Portal comes in two variations, both of which are included and fully supported in IBM Tivoli Monitoring 6.1:

  – Desktop Client: Automatically installed with the TEP on Windows OS

  – Web Client: Java-based GUI that can be used on a Windows on Linux OS

- Tivoli Enterprise Portal Server (TEPS)

  The Tivoli Enterprise Portal Server functions as a repository for user data, such as the user IDs and user access control for the monitoring data, meaning what data each user will be able to access and how it is displayed. It connects to the Hub TEMS and is accessed by the TEP clients. The TEPS acts as a logical gateway to the data managed by the TEMS. The Tivoli Enterprise Portal Server retrieves, manipulates, analyzes, and preformats data from the management server prior to passing it to the TEP.

  It uses a DB2 or Microsoft SQL database to store data.

- Tivoli Enterprise Monitoring Server (TEMS)

  The Tivoli Enterprise Monitoring Server is the key component on which all other architectural components depend, directly or indirectly. The TEMS acts as a control point and maintains a central repository of short-term data gathered from monitoring agents. It is where the definitions are stored for conditions that indicate a problem with a particular resource. The TEMS also controls the security for your monitoring solution. The Tivoli Enterprise Monitoring Server keeps information such as situation definitions (thresholds) and policies in a local datastore.

- Tivoli Data Warehouse (TDW)

  The current TEMS and Tivoli Enterprise Monitoring Agents (TEMAs) do not have the capability to directly write to a relational database (DB2, Oracle, or Microsoft SQL Server). They use an intermediate interface to store metrics collected from the enterprise. This “interface” is called the Warehouse Proxy agent.

  – Warehouse Proxy agent

    The Warehouse Proxy agent’s only function is to handle data warehousing requests from all managed systems in the enterprise. It uses an ODBC connection to write the historical data to a supported relational database.

This data can be used to populate custom reports (views and workspaces) in TEP or can be further analyzed using third-party reporting software.

The Warehouse Proxy agent can only connect to a Hub TEMS.

– Summarization and Pruning agent

  The Summarization and Pruning agent enables a customer to aggregate the raw data into hourly, daily, and weekly summarizations.

  It also accommodates the pruning of warehoused metrics that are no longer needed.

The bottom half of the diagram in Figure 6-2 on page 321 shows the IBM Tivoli Monitoring 5.1.2 environment:

- Tivoli management region server

  Computer on which Tivoli Management Framework software is installed; includes the libraries, binaries, data files, and the following user interfaces for installing and managing your Tivoli environment:

  – Tivoli Desktop: Traditional Tivoli graphical user interface (GUI)
  – Command-line interface (CLI): UNIX command line or the bash shell on a Windows operating system, from which you can perform system operations instead of using the Tivoli desktop

- IBM Tivoli Monitoring 5.1.2 monitoring engine

  Runs resource models and handles data logging and sending events. IBM Tivoli Monitoring 5.1.2 Fix Pack 6 is required.

- Endpoints

  Computers being monitored, provide communication and configuration of the monitoring system. The Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint resides on the same endpoint as the IBM Tivoli Monitoring 5.1.2 monitoring engine. It interacts directly with the IBM Tivoli Monitoring 5.1.2 monitoring engine on the endpoint via local pipe connection and either stores its historical data at the endpoint or at the TEMS.

- Gateways

  Control the communication and operations for endpoints.

- IBM Tivoli Data Warehouse 1.x (optional application)

  Optional and separate application that stores historical data.

- Web Health Console

  Web-based graphical interface included with the IBM Tivoli Monitoring software that enables you to view the health of resource models on an endpoint or to view real-time or historical data.

6.4 Why integrate IBM Tivoli Monitoring 5.1.2 with Version 6.1?

The intent of IBM Tivoli Monitoring 6.1 is not to replace the current IBM Tivoli Monitoring 5.x infrastructure. Several clients spent a lot of time and money developing a monitoring solution based on the previous IBM Tivoli Monitoring version and these environments also contains a lot of custom resource models.

IBM Tivoli Monitoring 6.1 represents an evolutionary next step in the IBM Tivoli Monitoring family that preserves the customer's investment in IBM Tivoli Monitoring 5.x and Candle OMEGAMON XE through simple interoperability and integration.

The main reason for such an integration is to enable IBM Tivoli Monitoring 5.1.2 to take advantage of the capabilities provided by the new TEPS/TEP and IBM Tivoli Data Warehouse components of IBM Tivoli Monitoring 6.1 solution.

6.4.1 TEPS/TEP

These components provide:

- New 6.1 Portal GUI capabilities, enabling a more complete and user-friendly interface for the performance and availability data
- Integration of multiple GUIs: IBM Tivoli Monitoring 5.x, Candle OMEGAMON/IBM Tivoli Monitoring 6.1, Candle OMEGAMON Z, IBM Tivoli Enterprise Console, and IBM Tivoli Data Warehouse
  - Reduced number of end-user clients to be installed and run in parallel
  - Quick link available for specific views
- Meaningful views and workspaces for real-time and long-term historical data
  - Trending with the IBM Tivoli Data Warehouse
  - Forensic data for problem reconstruction
  - Simplified schema for easier manipulation and extraction of data

6.4.2 IBM Tivoli Data Warehouse

One of the issues that currently causes concern with the IBM Tivoli Data Warehouse V1.x enablement is that it supports only aggregated data. This aggregation limits a customer's ability to use the data in support of diagnosis of their problems. The aggregation levels are fixed at one hour, but clients want the ability to maintain the raw data. By creating an IBM Tivoli Data Warehouse 2.1 / IBM Tivoli Monitoring 6.1 agent that receives data directly from the IBM Tivoli

Monitoring 5.1.2 monitoring engine, this data can then be inserted into the IBM Tivoli Data Warehouse 2.1 directly, without first aggregating that data.

The Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint uses the IRA framework to solve this issue in conjunction with delivering IBM Tivoli Monitoring 5.1.2 visualization in the Tivoli Enterprise Portal (TEP). The Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint will run at the IBM Tivoli Monitoring 5.1.2 endpoints, as opposed to with the gateway, to enable direct communication with the IBM Tivoli Monitoring 5.1.2 monitoring engine.

By leveraging the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint, the IBM Tivoli Monitoring 5.1.2 engine can forward data asynchronously, enabling the agent to collect and record the data to use both for real-time visualization (report situations) and for warehousing (historical situations).

The advantages of the IBM Tivoli Monitoring V5.x integration with IBM Tivoli Data Warehouse are:

- IBM Tivoli Data Warehouse 2.1 also stores instance data, which enormously improves the problem diagnosis capabilities inside the portal.

  Some clients are accessing the local IBM Tivoli Monitoring engine DB directly for that reason.
- Metric data from custom Resource Model can be included in the Warehouse much more easily, instead of writing a custom ETL for TDW 1.x.
- It is free of charge for IBM Tivoli Monitoring 5.x clients.

For more information about IBM Tivoli Data Warehouse 2.1, refer to Chapter 4, “Historical summarized data” on page 185.

6.5 Supported operating systems

Table 6-1 shows the operating systems supported at GA. On these systems, you can install the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint.

Table 6-1 Supported Operating Systems

Windows	UNIX
► Windows 2000 (Professional, Server, Advanced Server, 32-bit) ► Windows XP Professional 32-bit ► Windows Server 2003 (Standard, Enterprise, 32-bit)	► AIX 5.1, 5.2, 5.3 (32-bit and 64-bit) ► Solaris 8,9,10 (32-bit and 64-bit) ► HP-UX 11i (32-bit and 64-bit) ► Linux Intel – SLES 8,9 (32-bit) – RHEL AS/ES 2.1,3,4 (32-bit) ► Linux on zSeries – SLES 8,9 (31-bit and 64-bit) – RHEL AS 3,4 (31-bit and 64-bit)

6.6 IBM Tivoli Monitoring component software supported by Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint

You can use the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint to view data from the following IBM Tivoli Monitoring component software:

- IBM Tivoli Monitoring 5.1.2 Fix Pack 6
- IBM Tivoli Monitoring for Databases 5.1.0: DB2 Fix Pack 6
- IBM Tivoli Monitoring for Databases 5.1.0: Oracle Fix Pack 7
- IBM Tivoli Monitoring for Databases 5.1.1: Microsoft SQL Server Fix Pack 2
- IBM Tivoli Monitoring for Databases 5.1.2: Sybase ASE
- IBM Tivoli Monitoring for Databases 5.1.0: Informix® Fix Pack 3
- IBM Tivoli Monitoring for Applications 5.1.0: mySAP.com Feature Pack 6
- IBM Tivoli Monitoring for Applications 5.1.0: Siebel eBusiness Applications Fix Pack 3

- IBM Tivoli Monitoring for Messaging and Collaboration 5.1.1: Lotus® Domino® Fix Pack 6 or 7
- IBM Tivoli Monitoring for Messaging and Collaboration 5.1.2: Microsoft Exchange Server Fix Pack 1
- IBM Tivoli Monitoring for Web Infrastructure 5.1.2: WebSphere Application Server Fix Pack 2, IF 3
- IBM Tivoli Monitoring for Web Infrastructure 5.1.0: WebLogic Server
- IBM Tivoli Monitoring for Web Infrastructure 5.1.0: Apache HTTP Server
- IBM Tivoli Monitoring for Web Infrastructure 5.1.0: iPlanet™ Web Server
- IBM Tivoli Monitoring for Web Infrastructure 5.1.0: Internet Information Server
- IBM Tivoli Monitoring for Business Integration 5.1.1: WebSphere MQ
- IBM Tivoli Monitoring for Business Integration 5.1.0: WebSphere MQ Integrator
- IBM Tivoli Monitoring for Business Integration 5.1.1: WebSphere InterChange Server
- IBM Tivoli Monitoring for Microsoft .NET 5.1.2: Commerce Server
- IBM Tivoli Monitoring for Microsoft .NET 5.1.2: Content Management Server
- IBM Tivoli Monitoring for Microsoft .NET 5.1.2: BizTalk® Server
- IBM Tivoli Monitoring for Microsoft .NET 5.1.2: Host Integration Server
- IBM Tivoli Monitoring for Microsoft .NET 5.1.2: Internet Security and Acceleration Server 2000
- IBM Tivoli Monitoring for Microsoft .NET 5.1.2: Internet Security and Acceleration Server 2004
- IBM Tivoli Monitoring for Microsoft .NET 5.1.2: SharePoint® Portal Server
- IBM Tivoli Monitoring for Microsoft .NET 5.1.2: UDDI Services Server
- IBM Tivoli Monitoring for Virtual Servers 5.1.2: Citrix MetaFrame Access Suite
- IBM Tivoli Monitoring for Virtual Servers 5.1.2: VMWare ESX

Refer to “Appendix A. IBM Tivoli Monitoring component software supported by Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint” of *IBM Tivoli Monitoring: IBM Tivoli Monitoring 5.x Endpoint Agent User's Guide*, SC32-9490, for additional updates.

6.7 Installation and configuration of the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint

IBM Tivoli Monitoring: IBM Tivoli Monitoring 5.x Endpoint Agent User's Guide, SC32-9490, provides the following information about the requirements for the IBM Tivoli Monitoring 5.1.2 coexistence with IBM Tivoli Monitoring 6.1:

- Requirements for the monitoring agent: This section includes a complete list of all supported operating systems and software requirements.
- Installation images required to complete the installation.

6.7.1 Installation and configuration procedures

To use Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint, you must first install components that are part of the following base products as well as the monitoring agent:

- IBM Tivoli Monitoring 6.1
- IBM Tivoli Monitoring 5.1.2 Fix Pack 6
- IBM Tivoli Monitoring Component Services 5.1.1 Fix Pack 2 or V5.1.3

In addition to the information in this section, you must use the *IBM Tivoli Monitoring Installation and Setup Guide*, GC32-9407, to install and configure the IBM Tivoli Monitoring 6.1 components that the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint requires. The following documentation also contains additional installation and configuration information for some of the software relevant to this monitoring agent:

- IBM Tivoli Monitoring Passport Advantage® Readme First
- *IBM Tivoli Monitoring Administrator's Guide,* SC32-9408
- IBM Tivoli Monitoring 5.1.2 Fix Pack 6 Readme

All of these documents come online with the product.

Table 6-2 on page 329 contains a list of the procedures for installing and configuring the software for the IBM Tivoli Monitoring 6.1 base product and the components required for implementing the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint. The installation procedure can be summarized in 12 steps.

Table 6-2 Installation and configuration procedures

Software	Procedure
IBM Tivoli Monitoring 6.1 components	1. Install and seed the Tivoli Enterprise Monitoring Server (TEMS).
	2. Install the Tivoli Enterprise Portal Server (TEPS).
	3. Install the Tivoli Enterprise Portal (TEP).
	4. Install support for the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint on the monitoring server and portal server. (TEMS, TEPS)
	5. Install and configure Warehouse Proxy agent for Historical Data Collection (optional).
	6. Configure Historical Data Collection (optional).
IBM Tivoli Monitoring 5.1.2 components	7.Install the IBM Tivoli Monitoring 5.1.2 Fix Pack 6 onto your system.
	8. Install IBM Tivoli Monitoring Component Services if it is not already installed.
	9. Reconfigure data logging to stop the data logging on the existing IBM Tivoli Monitoring 5.1, 5.1.1, or 5.1.2 system (optional).
Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint	10. Install the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint using the **`winstall`** command.
	11. Configure IBM Tivoli Monitoring and the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint.
IBM Tivoli Monitoring 6.1 components	12. Start the Tivoli Enterprise Portal to verify your installation.

6.7.2 IBM Tivoli Monitoring 6.1 components (procedures 1-6)

Procedures 1-6 are related to the installation and the customization of the base IBM Tivoli Monitoring 6.1 components:

1. TEMS
2. TEPS
3. TEP
4. Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint support for TEMS and TEPS

5. Warehouse Proxy agent
6. Historical data collection configuration

For more information about procedures 1 through 4, refer to Chapter 3, “IBM Tivoli Monitoring 6.1 installation and first customization” on page 55.

Similarly, for procedures 5 and 6, you can refer to Chapter 4, “Historical summarized data” on page 185.

6.7.3 IBM Tivoli Monitoring 5.1.2 components (procedures 7-9)

Procedure 7. Installing IBM Tivoli Monitoring 5.1.2 Fix Pack 6

This Fix Pack provides full integration with IBM Tivoli Monitoring 6.1

Prerequisite of this Fix Pack is the base IBM Tivoli Monitoring 5.1.2 installation.

To install the patch, you can use either the Tivoli Desktop or the **`wpatch`** CLI.

Refer to the instructions detailed in the Fix Pack readme file for the exact steps to follow during the installation. It can be found at:

`ftp://ftp.software.ibm.com/software/tivoli_support/patches/patches_5.1.2/5.1.2-ITM-FP06/5.1.2-ITM-FP06.README`

Procedure 8. Installing IBM Tivoli Monitoring Component Services

The Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint uses common features of IBM Tivoli Monitoring Component Services to install and configure this agent.

If you are still using IBM Tivoli Monitoring Component Services in your environment, make sure the version you have is one of the following:

- V5.1.1 Fix Pack 2
  To download the Fix Pack package, refer to:

  `ftp://ftp.software.ibm.com/software/tivoli_support/patches/patches_5.1.1/5.1.1-ICS-FP02/`

  or

- V5.1.3
  To download the base package for V5.1.3, refer to:

  `ftp://ftp.software.ibm.com/software/tivoli_support/patches/patches_5.1.3/5.1.3-TIV-ITM_ICS-0000/`

To install the Fix Pack or the base product, you can use either the Tivoli Desktop or the **wpatch** CLI.

Refer to the instructions detailed in the Fix Pack or product documentation for the exact steps to follow.

Procedure 9. Reconfiguring data logging

This optional procedure enables you to determine which kind of data you can collect from your Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint.

A new IBM Tivoli Monitoring CLI can be used to perform this configuration:

```
wdmepconfig
```

You can configure the seeding information to let the agent log data of ITM5, ITM6, or both with the following result:

- ITM5: The data will be written in the old IBM Tivoli Data Warehouse 1.x.
- ITM6: The data will be written in the new IBM Tivoli Data Warehouse 2.1.
- BOTH: The data will be written on IBM Tivoli Data Warehouse 1.x *and* 2.1.

For example:

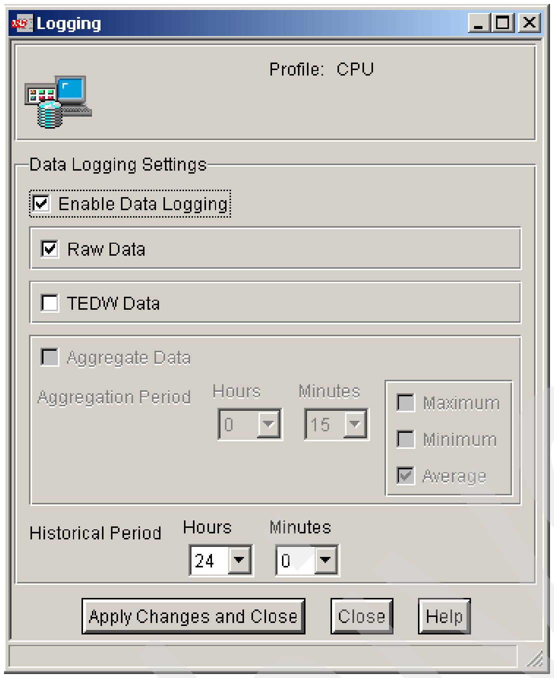
```
wdmepconfig -e endpoint -D DataSeeding=ITM6
```

If you choose to collect data only for ITM6, you are asked to stop the rollup of the .xml files that contain the aggregated warehouse data from the endpoints to the RIM, and change the IBM Tivoli Monitoring profile collection options to disable the IBM Tivoli Data Warehouse logging.

To stop the collection, you can use this command:

```
wdmcollect -m all -t
```

To change the Tmw2kProfile Logging configuration, deselect the TEDW Data check box as shown in Figure 6-3 on page 332.


Logging

Profile: CPU

Data Logging Settings

[✓] Enable Data Logging

[✓] Raw Data

[ ] TEDW Data

[ ] Aggregate Data

Aggregation Period — Hours: 0 — Minutes: 15

[ ] Maximum

[ ] Minimum

[✓] Average

Historical Period — Hours: 24 — Minutes: 0

Figure 6-3 Data Logging properties for a Resource Model

Note: The Enable Data Logging selection does not prevent the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint integration in the TEP, but there is a difference in the way the data is displayed:

- **Enable Data Logging** unchecked

  Only the Resource Model Health data is available in the TEP. The Health data provides the same information obtained with the **`wdmlseng`** command.

- **Enable Data Logging** checked

  – If no other selections are made, it behaves exactly as if unchecked.

  – Regardless of RAW Data or **Aggregated Data** selection, only RAW Data will be sent to the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint engine, ktmcma, for processing, so you will be able to see the logging data in the TEP.

    Aggregated data only applies to the IBM Tivoli Monitoring 5.x database.

6.7.4 Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint (procedures 10-11)

Procedure 10. Installing the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint

Installation of this component requires IBM Tivoli Monitoring 5.1.2 Fix Pack 6, which enables full compatibility between the IBM Tivoli Monitoring 5.1.2 Agent and the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint.

The installation can be performed using either **`winstall`** or the Tivoli Desktop.

- Desktop

  To install using the Tivoli Desktop, complete the following steps:

  a. Select **Install** → **Install Product** from the Desktop menu.

  > **Note:** If the last installation directory is no longer current, you could see a harmless error message about the media settings. Just click **OK**.

  b. Click **Select Media** to display the File Browser window and locate the directory containing the ITM61AGT.IND file, then click **Set Path** → **Set Media and Close**.

  c. This opens the Install Product dialog shown in Figure 6-4 on page 334.

Select the software name from the Select Product to Install list. Select the targets of your installation that should be present in the Clients to Install On list.

Note: The ITM61AGT product must be installed on the TMR server and all Gateways in your environment that have Tmw2kProfile distributed and for which you want to install the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint code.

If your client is listed in the Available Clients list no action will be taken, and it could mean that the ITM61AGT code has been already installed on that client.

Click **Install & Close** when you have completed the Clients to Install On selection.

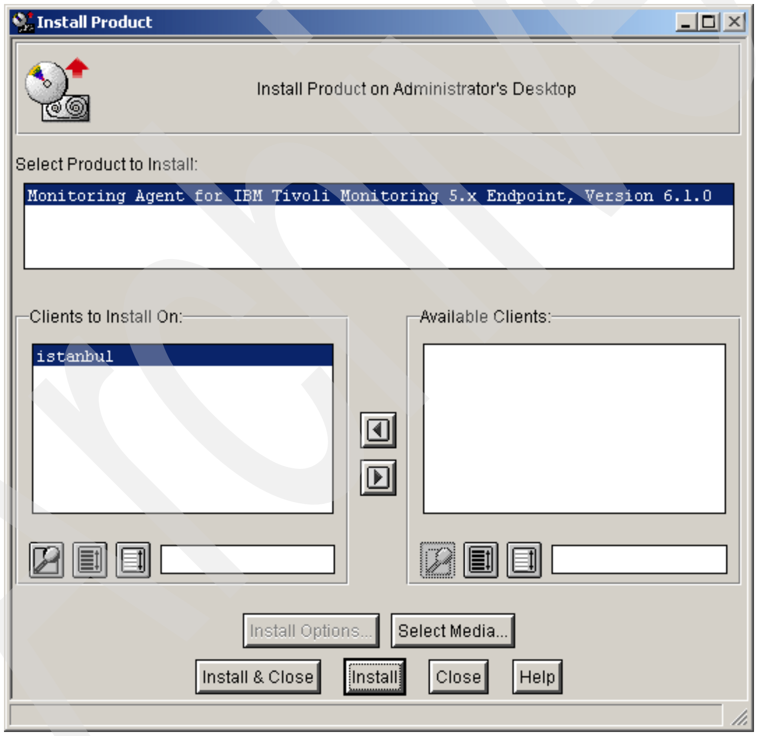
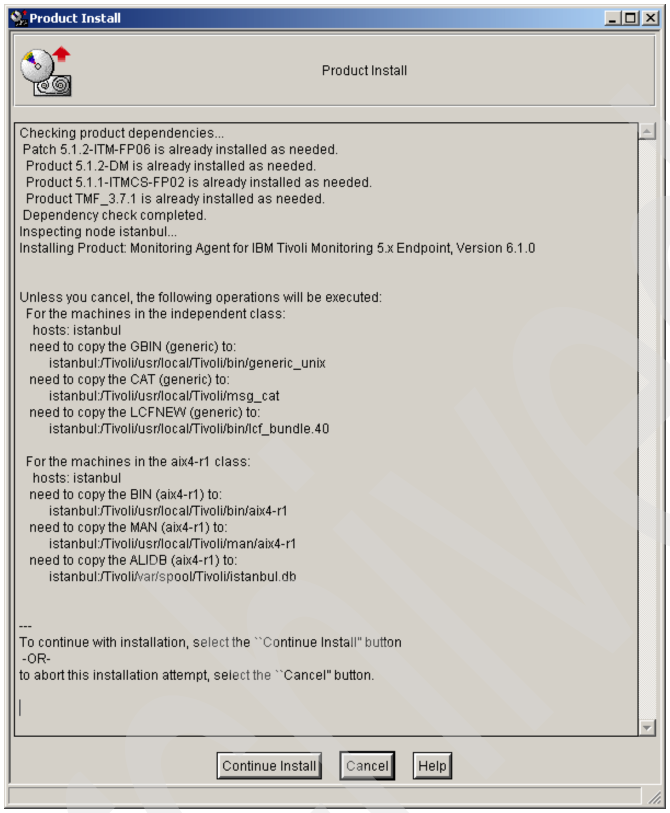

Install Product
Install Product on Administrator's Desktop
Select Product to Install:
Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint, Version 6.1.0
Clients to Install On:
istanbul
Available Clients:
Install Options...
Select Media...
Install & Close
Install
Close
Help

Figure 6-4 Install Product dialog

d. The Product Install dialog shown in Figure 6-5 is displayed. Click **Continue Install** to start the installation process.

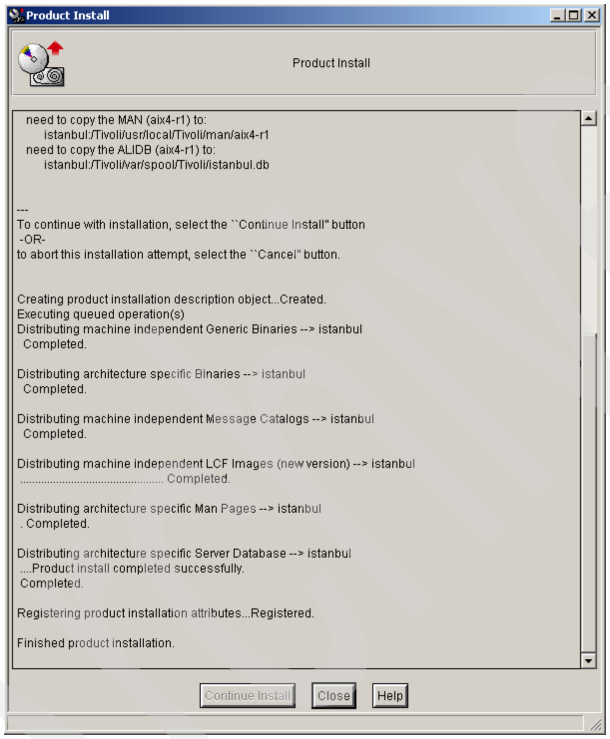


```
Checking product dependencies...
 Patch 5.1.2-ITM-FP06 is already installed as needed.
  Product 5.1.2-DM is already installed as needed.
  Product 5.1.1-ITMCS-FP02 is already installed as needed.
  Product TMF_3.7.1 is already installed as needed.
 Dependency check completed.
Inspecting node istanbul...
Installing Product: Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint, Version 6.1.0


Unless you cancel, the following operations will be executed:
  For the machines in the independent class:
    hosts: istanbul
   need to copy the GBIN (generic) to:
        istanbul:/Tivoli/usr/local/Tivoli/bin/generic_unix
   need to copy the CAT (generic) to:
        istanbul:/Tivoli/usr/local/Tivoli/msg_cat
   need to copy the LCFNEW (generic) to:
        istanbul:/Tivoli/usr/local/Tivoli/bin/lcf_bundle.40

  For the machines in the aix4-r1 class:
    hosts: istanbul
   need to copy the BIN (aix4-r1) to:
        istanbul:/Tivoli/usr/local/Tivoli/bin/aix4-r1
   need to copy the MAN (aix4-r1) to:
        istanbul:/Tivoli/usr/local/Tivoli/man/aix4-r1
   need to copy the ALIDB (aix4-r1) to:
        istanbul:/Tivoli/var/spool/Tivoli/istanbul.db


---
To continue with installation, select the ``Continue Install'' button
 -OR-
to abort this installation attempt, select the ``Cancel'' button.
```

Figure 6-5 Product Install dialog

e. When the installation is complete, a message at the end of the Product Install dialog says `Finished product installation`. Figure 6-6 shows the sample dialog.

> **Note:** Scroll up the page to check that all components show `Completed` status. In case of failure during the installation, do not proceed with the next steps until you solve the installation issue or have contacted Tivoli Customer Support.



```
  need to copy the MAN (aix4-r1) to:
      istanbul:/Tivoli/usr/local/Tivoli/man/aix4-r1
  need to copy the ALIDB (aix4-r1) to:
      istanbul:/Tivoli/var/spool/Tivoli/istanbul.db


---
To continue with installation, select the ``Continue Install" button
 -OR-
to abort this installation attempt, select the ``Cancel" button.


Creating product installation description object...Created.
Executing queued operation(s)
Distributing machine independent Generic Binaries --> istanbul
  Completed.

Distributing architecture specific Binaries --> istanbul
  Completed.

Distributing machine independent Message Catalogs --> istanbul
  Completed.

Distributing machine independent LCF Images (new version) --> istanbul
 ............................................ Completed.

Distributing architecture specific Man Pages --> istanbul
 . Completed.

Distributing architecture specific Server Database --> istanbul
 ....Product install completed successfully.
 Completed.

Registering product installation attributes...Registered.

Finished product installation.
```

Figure 6-6 Product Install complete dialog

- Command line interface: **`winstall`**

  To install using the command line interface, run the following command from the command prompt:

  ```
  winstall -c source_dir [-s server] -i product [-y] [-n | managed_node...]
  ```

  In this sequence:

  `-c` *source_dir*

  Specifies the complete path to the directory containing the installation image.

  `-s` *server*

  Specifies the name of the managed node to use as the installation server. By default, the installation server is the Tivoli server.

  `-i` *product*

  Specifies the product index file from which the product is installed. (ITM61AGT.IND is the file; you can omit the .IND in the command)

  `-y`

  Installs the product without requesting confirmation.

  `-n`

  Installs the product on all managed nodes that do not have the product installed. This option is ignored when *managed_node* is specified.

  `managed_node`

  Specifies the managed nodes on which to install this Tivoli Enterprise product. You can specify multiple managed nodes. If you do not specify a managed node, the product is installed on all managed nodes in your Tivoli region.

Refer to *Tivoli Management Framework Reference Manual, Version 4.1.1*, SC32-0806, for more information about this command.

Procedure 11. Configuring and distributing the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint

This is the last step you must perform in order to enable data collection from the managed systems that you want to monitor.

You can perform this procedure in two ways:

- Distribution
- Manual distribution

If you have sufficient bandwidth to transfer all of the files, use the distribution procedure. If your bandwidth is not sufficient, manually distribute the files and push only the configuration.

The Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint is started and stopped by the IBM Tivoli Monitoring 51.2 engine and installed in the IBM Tivoli Monitoring 51.2 environment. To facilitate installation, configuration, and operation, it is installed using traditional Tivoli endpoint distribution mechanisms through the new **witm61agt** command. Refer to 6.15, "Monitoring Agent behind the scenes" on page 393 for additional details about this Agent.

You can also use this command to modify the configuration of an endpoint where the agent is already running. The changes specified by using this procedure take effect the next time the engine is restarted.

Distribution

1. Use the following command on each targeted system you have listed;

```
witm61agt -c CMSAddress[:BackupCMSAddress] [-f] [-n] [-i seeding] [-o
outfile] [[-D Variable=Value] ... ] [ -P protocol ] [-p port] {-a |
endpoint [endpoint ...] | @filename}
```

   Where:

-f	Force the distribution to proceed even if the operating system version check fails.
-n	No distribution of binaries, just perform configuration.
-r	Removes files from endpoints, rather than distributing.
-i	IBM Tivoli Monitoring **wdmepconfig** command will also be invoked on each endpoint. `seeding` specifies where data will go. Valid values are ITM5, ITM6, or BOTH.
-o	Does not distribute binaries or configure, just dumps endpoint list to `outfile`.
-D	Adds the setting `Variable=Value` to the environment file for the agent.
-P	Specifies the network protocol to use: TCP or UDP. Default is TCP.
-c	Specifies the network name of the management server. Optionally, a backup server can be specified, separated by a colon (:).
-p	Specifies the TCP/IP port number on which the management server listens.

-a	All endpoints that have IBM Tivoli Monitoring profiles distributed to them will receive the distribution.
endpoint	Distributes to the named endpoint or endpoints.
@filename	Distributes to all endpoints named in the specified file.

2. If you have not set the logging behavior of the IBM Tivoli Monitoring engine using the -i option in the previous step, you can use the following command, which sets only that logging behavior:

```
wdmepconfig {-e endpoint | @endpoint_file} {-D DataSeeding=ITM5 | ITM6 |
BOTH}
```

Where:

`-e` *endpoint*

For IBM Tivoli Monitoring 5.1.2, a list of one or more names of the endpoints

`-e` *@endpoints_file*

For IBM Tivoli Monitoring 5.1.2, the file that contains the list of endpoints

`-D DataSeeding`

Application to which to log data; possible values:

ITM5	IBM Tivoli Monitoring 5.1.2. This is the default value. The data is logged in the Tivoli Data Warehouse and the Web Health Console, but not the Tivoli Enterprise Portal.
ITM6	IBM Tivoli Monitoring 6.1. The data is logged in the Tivoli Enterprise Portal, but not in the Tivoli Data Warehouse and the Web Health Console.
BOTH	IBM Tivoli Monitoring 5.1.2 and IBM Tivoli Monitoring 6.1. This is the preferred setting. You can start with this option, decide which one is preferred, and then reconfigure accordingly.

Note: Both commands **`wdmepconfig`** and the **`witm61agent`** (if the -i flag is specified, during its execution it will recall the **`wdmepconfig`** command) generate a mdist2 distribution. Make sure you previously configured the mdist2 database and you can successfully open your Mdist2 GUI from the Tivoli Desktop before executing any of these commands.

If the mdist2 RIM object is already present on your Tivoli management region server, you can use this command to check the connectivity to the Relational Database:

```
wrimtest -l mdist2
```

If the connection is successful you will get an output like the following:

```
[root@istanbul][/]-> wrimtest -l mdist2
Resource Type  : RIM
Resource Label : mdist2
Host Name      : istanbul
User Name      : db2inst1
Vendor         : DB2
Database       : MDIST2
Database Home  : /usr/opt/db2_08_01
Server ID      : tcpip
Instance Home  : /home/db2inst1
Instance Name  : db2inst1
Opening Regular Session...Session Opened
RIM : Enter Option >
```

Simply press x to return to the prompt.

Example 6-1 shows an implementation made in our lab environment.

Example 6-1 Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint Distribution

```
In the following example, the agent is distributed to all endpoints for both
versions of IBM TIvoli Monitoring on the myTEMS host (hostname of the myTEMS is
mytems) using the default port 1918:

witm61agt -i BOTH -c mytems -a

In the following example, the agent is distributed on endpoint fab for IBM
TIvoli Monitoring 6.1 only on the myTEMS host (hostname mytems) and
mybackupTEMS host (hostname mybackuptems):

witm61agt fab-ep -i ITM6 -c mytems:mybackuptems

The output generated by this command looks like the following:
```

```
[root@istanbul][/opt/Tivoli/lcf_fab/dat/2]-> witm61agt fab-ep -i ITM6 -c
mytems:mybackuptems
KTM0009I Endpoint fab-ep: SUCCEEDED
KTM0030I Performing IBM Tivoli Monitoring engine configuration on all
endpoints.

set parameter command successfully submitted to mdist2 with timeout 300, the
distribution ID is 1567175343.158.
.
        Operation completed successfully on endpoint fab-ep

Command completed: operation completed successfully on 1 endpoints,
unsuccessfully on 0 endpoints, remaining 0.
```

You can also check the result of the distribution on the mdist2 database using the wmdist command:

```
[root@istanbul][/]-> wmdist -e 158
Name                  Status         Start Time           End Time
istanbul              SUCCESSFUL     2005.10.14 17:02:58  2005.10.14 17:02:59
fab-ep                SUCCESSFUL     2005.10.14 17:02:58  2005.10.14 17:02:59
```

The installation of the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint code creates three dependency set on the TMR server, that are used during the **witm61agt command** to install the agent code.
You can check the dependency creation using these command:

wlookup -ar DependencyMgr |grep ktm

This command will list the dependency names and their OIDs:

```
[root@istanbul][/]-> wlookup -ar DependencyMgr|grep ktm
ktm_init        1567175343.1.1449#Depends::Mgr#
ktm_init296     1567175343.1.1450#Depends::Mgr#
ktm_metadata    1567175343.1.1448#Depends::Mgr#
```

The ktm_init296 contains the dependencies for gcc-2.96 based Linux systems, for example RedHat ES 2.1, ktm_init for all the other Operating Systems.

wdepset -v *dep_OID*

This command will list all the files associated to the dependency:

```
[root@istanbul][/]-> wdepset -v 1567175343.1.1449#Depends::Mgr#
aix4-r1:
        bin/aix4-r1/TME/KTM/ktmcma,../../bin/aix4-r1/TME/KTM,8
        bin/aix4-r1/TME/KTM/kbbacf1,../../bin/aix4-r1/TME/KTM,8
        lib/aix4-r1/TME/KTM/libkbb.a,../../lib/aix4-r1,8
        lib/aix4-r1/TME/KTM/libkra.a,../../lib/aix4-r1,8
        lib/aix4-r1/TME/KTM/libkhdxcl1.a,../../lib/aix4-r1,8
        lib/aix4-r1/TME/KTM/libkdc.a,../../lib/aix4-r1,8
        lib/aix4-r1/TME/KTM/libkglbase.a,../../lib/aix4-r1,8
        lib/aix4-r1/TME/KTM/libkdsfilt.a,../../lib/aix4-r1,8
        lib/aix4-r1/TME/KTM/libknsnls2.a,../../lib/aix4-r1,8
        lib/aix4-r1/TME/KTM/libkgl01p1.a,../../lib/aix4-r1,8
        lib/aix4-r1/TME/KTM/libkgl01p2.a,../../lib/aix4-r1,8
        lib/aix4-r1/TME/KTM/libkde.a,../../lib/aix4-r1,8
        lib/aix4-r1/TME/KTM/libkdh.a,../../lib/aix4-r1,8
        lib/aix4-r1/TME/KTM/libkns.a,../../lib/aix4-r1,8
        lib/aix4-r1/TME/KTM/libkcutu32.a,../../lib/aix4-r1,8
        lib/aix4-r1/TME/KTM/libkcui18n32.a,../../lib/aix4-r1,8
        lib/aix4-r1/TME/KTM/libkcuuc32.a,../../lib/aix4-r1,8
        lib/aix4-r1/TME/KTM/libkcudt32.a,../../lib/aix4-r1,8
        lib/aix4-r1/TME/KTM/libkcuio32.a,../../lib/aix4-r1,8
        lib/aix4-r1/TME/KTM/libkcutu.a,../../lib/aix4-r1,8
        lib/aix4-r1/TME/KTM/libkcui18n.a,../../lib/aix4-r1,8
        lib/aix4-r1/TME/KTM/libkcuuc.a,../../lib/aix4-r1,8
        lib/aix4-r1/TME/KTM/libkcudt.a,../../lib/aix4-r1,8
        lib/aix4-r1/TME/KTM/libkcuio.a,../../lib/aix4-r1,8
...
...
depset:
        1567175343.1.1448#Depends::Mgr#

There is another depset associated to this dependency that is ktm_metadata that
is distributed to all the interpreter types (generic):

[root@istanbul][/]-> wdepset -v 1567175343.1.1448#Depends::Mgr#
generic:
        generic/TME/KTM/ATTRLIB/kib.atr,LCFNEW/KTM/ATTRLIB,8
        generic/TME/KTM/ATTRLIB/ktm.atr,LCFNEW/KTM/ATTRLIB,8
        generic/TME/KTM/ATTRLIB/ktm_aba.atr,LCFNEW/KTM/ATTRLIB,8
        generic/TME/KTM/ATTRLIB/ktm_abh.atr,LCFNEW/KTM/ATTRLIB,8
        generic/TME/KTM/ATTRLIB/ktm_ama.atr,LCFNEW/KTM/ATTRLIB,8
        generic/TME/KTM/ATTRLIB/ktm_amb.atr,LCFNEW/KTM/ATTRLIB,8
        generic/TME/KTM/ATTRLIB/ktm_amd.atr,LCFNEW/KTM/ATTRLIB,8
        generic/TME/KTM/ATTRLIB/ktm_amn.atr,LCFNEW/KTM/ATTRLIB,8
        generic/TME/KTM/ATTRLIB/ktm_ams.atr,LCFNEW/KTM/ATTRLIB,8
        generic/TME/KTM/ATTRLIB/ktm_amw.atr,LCFNEW/KTM/ATTRLIB,8
        generic/TME/KTM/ATTRLIB/ktm_amx.atr,LCFNEW/KTM/ATTRLIB,8
        generic/TME/KTM/ATTRLIB/ktm_biw.atr,LCFNEW/KTM/ATTRLIB,8
        generic/TME/KTM/ATTRLIB/ktm_bix.atr,LCFNEW/KTM/ATTRLIB,8
        generic/TME/KTM/ATTRLIB/ktm_ctd.atr,LCFNEW/KTM/ATTRLIB,8
        generic/TME/KTM/ATTRLIB/ktm_cto.atr,LCFNEW/KTM/ATTRLIB,8
        generic/TME/KTM/ATTRLIB/ktm_ctq.atr,LCFNEW/KTM/ATTRLIB,8
        generic/TME/KTM/ATTRLIB/ktm_ctr.atr,LCFNEW/KTM/ATTRLIB,8
        generic/TME/KTM/ATTRLIB/ktm_ctw.atr,LCFNEW/KTM/ATTRLIB,8
        generic/TME/KTM/ATTRLIB/ktm_cty.atr,LCFNEW/KTM/ATTRLIB,8
        generic/TME/KTM/ATTRLIB/ktm_gms.atr,LCFNEW/KTM/ATTRLIB,8
        generic/TME/KTM/ATTRLIB/ktm_gwa.atr,LCFNEW/KTM/ATTRLIB,8
        generic/TME/KTM/ATTRLIB/ktm_gwi.atr,LCFNEW/KTM/ATTRLIB,8
        generic/TME/KTM/ATTRLIB/ktm_gwl.atr,LCFNEW/KTM/ATTRLIB,8
        generic/TME/KTM/ATTRLIB/ktm_gwp.atr,LCFNEW/KTM/ATTRLIB,8
        generic/TME/KTM/ATTRLIB/ktm_iqs.atr,LCFNEW/KTM/ATTRLIB,8
        generic/TME/KTM/ATTRLIB/ktm_iqy.atr,LCFNEW/KTM/ATTRLIB,8
        generic/TME/KTM/ATTRLIB/ktm_iqz.atr,LCFNEW/KTM/ATTRLIB,8
        generic/TME/KTM/ATTRLIB/ktm_iud.atr,LCFNEW/KTM/ATTRLIB,8
        generic/TME/KTM/ATTRLIB/ktm_iui.atr,LCFNEW/KTM/ATTRLIB,8
        generic/TME/KTM/ATTRLIB/ktm_ivd.atr,LCFNEW/KTM/ATTRLIB,8
        generic/TME/KTM/ATTRLIB/ktm_ivi.atr,LCFNEW/KTM/ATTRLIB,8
        generic/TME/KTM/ATTRLIB/ktm_ixa.atr,LCFNEW/KTM/ATTRLIB,8
        generic/TME/KTM/ATTRLIB/ktm_ixb.atr,LCFNEW/KTM/ATTRLIB,8
        generic/TME/KTM/ATTRLIB/ktm_ixt.atr,LCFNEW/KTM/ATTRLIB,8
        generic/TME/KTM/ATTRLIB/ktm_iym.atr,LCFNEW/KTM/ATTRLIB,8
        generic/TME/KTM/ATTRLIB/ktm_izy.atr,LCFNEW/KTM/ATTRLIB,8
        generic/TME/KTM/ATTRLIB/ktm_mmi.atr,LCFNEW/KTM/ATTRLIB,8
        generic/TME/KTM/metadata/Active_Directory_DC.xml,LCFNEW/KTM/metadata,8

generic/TME/KTM/metadata/Active_Directory_Replication.xml,LCFNEW/KTM/metadata,8
        generic/TME/KTM/metadata/Apache.xml,LCFNEW/KTM/metadata,8
        generic/TME/KTM/metadata/CitrixMetaFrameAS.xml,LCFNEW/KTM/metadata,8
        generic/TME/KTM/metadata/DB2Monitoring.xml,LCFNEW/KTM/metadata,8
        generic/TME/KTM/metadata/DHCP.xml,LCFNEW/KTM/metadata,8
        generic/TME/KTM/metadata/DNS.xml,LCFNEW/KTM/metadata,8
        generic/TME/KTM/metadata/Domino.xml,LCFNEW/KTM/metadata,8
        generic/TME/KTM/metadata/IBMInformix.xml,LCFNEW/KTM/metadata,8
        generic/TME/KTM/metadata/IIS.xml,LCFNEW/KTM/metadata,8
        generic/TME/KTM/metadata/IPlanet.xml,LCFNEW/KTM/metadata,8

generic/TME/KTM/metadata/Microsoft_Exchange_Server.xml,LCFNEW/KTM/metadata,8
        generic/TME/KTM/metadata/Microsoft_SQL_Server.xml,LCFNEW/KTM/metadata,8
        generic/TME/KTM/metadata/MQ_Workflow.xml,LCFNEW/KTM/metadata,8
        generic/TME/KTM/metadata/MSBizMon.xml,LCFNEW/KTM/metadata,8
        generic/TME/KTM/metadata/MSCMMon.xml,LCFNEW/KTM/metadata,8
        generic/TME/KTM/metadata/MSCSMon.xml,LCFNEW/KTM/metadata,8
        generic/TME/KTM/metadata/MSHISMon.xml,LCFNEW/KTM/metadata,8
        generic/TME/KTM/metadata/MSISA04Mon.xml,LCFNEW/KTM/metadata,8
        generic/TME/KTM/metadata/MSISAMon.xml,LCFNEW/KTM/metadata,8
        generic/TME/KTM/metadata/MSSPPMon.xml,LCFNEW/KTM/metadata,8
        generic/TME/KTM/metadata/MSUDDIMon.xml,LCFNEW/KTM/metadata,8
        generic/TME/KTM/metadata/mySAP.xml,LCFNEW/KTM/metadata,8
        generic/TME/KTM/metadata/Oracle.xml,LCFNEW/KTM/metadata,8
        generic/TME/KTM/metadata/Resource_Model.xml,LCFNEW/KTM/metadata,8
        generic/TME/KTM/metadata/Siebel.xml,LCFNEW/KTM/metadata,8
        generic/TME/KTM/metadata/Solaris.xml,LCFNEW/KTM/metadata,8
        generic/TME/KTM/metadata/SybaseMon.xml,LCFNEW/KTM/metadata,8
        generic/TME/KTM/metadata/UNIX_Linux.xml,LCFNEW/KTM/metadata,8
```

```
    generic/TME/KTM/metadata/VMWareMon.xml,LCFNEW/KTM/metadata,8
    generic/TME/KTM/metadata/Weblogic.xml,LCFNEW/KTM/metadata,8
    generic/TME/KTM/metadata/WebsphereAS.xml,LCFNEW/KTM/metadata,8
    generic/TME/KTM/metadata/Websphere_MQ.xml,LCFNEW/KTM/metadata,8
    generic/TME/KTM/metadata/Websphere_MQI.xml,LCFNEW/KTM/metadata,8
    generic/TME/KTM/metadata/WICS.xml,LCFNEW/KTM/metadata,8
    generic/TME/KTM/metadata/Windows.xml,LCFNEW/KTM/metadata,8

The code is about 45MB is size. Differentiating the output of the Index.v5 file
on the $LCF_DATDIR directory of the Tivoli Management Agent before and after
the witm61agt command execution, you will see that all the above files have
been added on the machine.
```

Manual distribution

Use the following command to manually copy the files on your endpoint and push only the configuration for both ITM5 and ITM6 data logging on a TEMS called mytems that uses the default port 1918:

```
witm61agt endpoint -n -i BOTH -c mytems
```

The -n option is for no dependency push.

Now distribute the code as follows:

1. Insert the CD on the computer with the endpoint.
2. Install the agent code, which is located in the /AGENT directory on the CD:
   a. Open the top-level directory of the lcf install directory where the endpoint is installed; for example: $LCFROOT.
   b. Extract the operating system-specific binary files from the /AGENT directory. Refer to Table 6-3 for the appropriate file and estimated size on the disk.

Table 6-3 File names for operating systems

Operating system	File name	Estimated size on disk
AIX	aix4-r1.tar.Z	~35 MB
HP/UX	hpux10.tar.Z	~40 MB
gcc-2.96 based Linux systems (for example, RedHat ES 2.1)	linux-ix86-296.tar.Z	~32 MB
gcc-3.x based Linux systems (all other supported Linux systems)	linux-ix86.tar.Z	~31 MB

Operating system	File name	Estimated size on disk
Linux on z/Series	linux-s390.tar.Z	~33 MB
Solaris	solaris2.tar.Z	~40 MB
Windows	w32-ix86.zip	~17 MB

3. Extract the selected files using the following command:

```
[root@istanbul][/]-> cd /opt/Tivoli/lcf/dat/1
[root@istanbul][/opt/Tivoli/lcf/dat/1]-> . ./lcf_env.sh
[root@istanbul][/opt/Tivoli/lcf/dat/1]-> cd $LCFROOT
[root@istanbul][/opt/Tivoli/lcf]-> zcat <filename> | tar xvf -
```

   where `<filename>` is file name for the operating system in Table 6-3 on page 344.

4. Extract the following metadata files from the $LCF_DATDIR/LCFNEW/KTM directory (created by the **witm61agt** command when it was run to configure the endpoint, even if the -n option was used so that the main bulk of code was not distributed):

   For UNIX systems: metadata.tar.Z

   For Windows systems: metadata.zip

Important: Regardless of the procedure used to install the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint code (Distribution or Manual distribution), recycle the IBM Tivoli Monitoring engine on each endpoint where you installed the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint code to enable the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint using either one of these commands:

```
wdmcmd -stop -e endpoint_label

wdmcmd -restart -e endpoint_label
```

Or simply use:

```
wdmcmd -restart -e endpoint_label
```

This starts a process called ktmcma spawned by the IBM Tivoli Monitoring engine itself.

For UNIX OS you can **grep** for the name of the process. The parent will be the Java process that represents the IBM Tivoli Monitoring engine.

```
[root@istanbul][/]-> ps -ef|grep ktmcma
    root 33374 38592   0 14:40:48      -  0:01
/opt/Tivoli/lcf/bin/aix4-r1/mrt/../TME/KTM/ktmcma
    root 39716 37902   0 14:43:10  pts/5  0:00 grep ktmcma
```

If you restart the IBM Tivoli Monitoring engine and you quickly run the above **ps** command you could notice multiple ktmcma processes, some of them child of 1. The ktmcma process shuts down gracefully when it reads EOF from the IBM Tivoli Monitoring engine, so it could take a few seconds if the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint Agent is doing something, such as uploading data. Only the process with the IBM Tivoli Monitoring engine as parent will stay running; the others, if there, will disappear shortly.

Example 6-2 shows an example of an implementation made on our environment.

Example 6-2 Manual distribution of Monitoring Agent

```
The following command shows a distribution

[root@istanbul][/opt/Tivoli/lcf_fab6]-> witm61agt fab6-ep -n -i ITM6 -c paris
KTM0009I Endpoint fab6-ep: SUCCEEDED
KTM0030I Performing IBM Tivoli Monitoring engine configuration on all
endpoints.

set parameter command successfully submitted to mdist2 with timeout 300, the
distribution ID is 1567175343.171.
.
```

```
        Operation completed successfully on endpoint fab5-ep

Command completed: operation completed successfully on 1 endpoints,
unsuccessfully on 0 endpoints, remaining 0.
```

The **-n** option is for no dependency push. If specified, the command will do the configuration but not push the dependencies. The following files and directories will be crated:

```
[root@istanbul][/opt/Tivoli/lcf_fab6/dat/8/LCFNEW]-> ls -laR
total 4
drwxrwxrwx   4 root     system         512 Oct 14 19:50 .
drwxr-xr-x   5 root     system         512 Oct 14 19:50 ..
drwxrwxrwx   4 root     system         512 Oct 14 19:49 KTM
drwxr-xr-x   3 root     system         512 Oct 14 19:50 Tmw2k

./KTM:
total 7
drwxrwxrwx   4 root     system         512 Oct 14 19:49 .
drwxrwxrwx   4 root     system         512 Oct 14 19:50 ..
-rw-r--r--   1 root     system        1080 Oct 14 19:50 KTMENV
drwxrwxrwx   2 root     system         512 Oct 14 19:49 hist
drwxrwxrwx   2 root     system         512 Oct 14 19:49 logs

./KTM/hist:
total 2
drwxrwxrwx   2 root     system         512 Oct 14 19:49 .
drwxrwxrwx   4 root     system         512 Oct 14 19:49 ..

./KTM/logs:
total 2
drwxrwxrwx   2 root     system         512 Oct 14 19:49 .
drwxrwxrwx   4 root     system         512 Oct 14 19:49 ..

./Tmw2k:
total 3
drwxr-xr-x   3 root     system         512 Oct 14 19:50 .
drwxrwxrwx   4 root     system         512 Oct 14 19:50 ..
drwxr-xr-x   4 root     system         512 Oct 14 19:50 Unix

./Tmw2k/Unix:
total 4
drwxr-xr-x   4 root     system         512 Oct 14 19:50 .
drwxr-xr-x   3 root     system         512 Oct 14 19:50 ..
drwxr-xr-x   2 root     system         512 Oct 14 19:50 bin
drwxr-xr-x   2 root     system         512 Oct 14 19:50 data

./Tmw2k/Unix/bin:
total 298
```

```
drwxr-xr-x   2 root     system          512 Oct 14 19:50 .
drwxr-xr-x   4 root     system          512 Oct 14 19:50 ..
-rwxr-xr-x   1 root     system         8131 Oct 14 19:50 getProfileInfo.sh
-rwxr-xr-x   1 root     system       140773 Oct 14 19:50 gzip

./Tmw2k/Unix/data:
total 4
drwxr-xr-x   2 root     system          512 Oct 14 19:50 .
drwxr-xr-x   4 root     system          512 Oct 14 19:50 ..
-rw-r--r--   1 root     system           19 Oct 14 19:50 tmw2k.config
-rwxr-xr-x   1 root     system           79 Oct 14 19:50 tmw2k_default.config

[root@istanbul][/opt/Tivoli/lcf_fab6/dat/8/LCFNEW]-> du -k
1       ./KTM/logs
1       ./KTM/hist
3       ./KTM
2       ./Tmw2k/Unix/data
149     ./Tmw2k/Unix/bin
151     ./Tmw2k/Unix
151     ./Tmw2k
155     .

Having a look to the Index.v5 file, you will see all the files downloaded by
the witm61agt command.

Index.v5 file content

1|0x434ac4d2|21123|00000000|False|4|14 Oct 2005 19:50:15|14 Oct 2005
19:49:53|/opt/Tivoli/lcf_fab6/dat/8/cache|/bin/aix4-r1/TME/KTM/ITM61Integration
_K
TMDaemon|
2|0x4335ecde|2665026|00000000|False|1|14 Oct 2005 19:50:30|14 Oct 2005
19:50:30|/opt/Tivoli/lcf_fab6/dat/8/cache|/bin/aix4-r1/TME/Tmw2k/tmw2k_ep|
3|0x4335dce2|79|00000000|True|1|14 Oct 2005 19:50:26|14 Oct 2005
19:50:26|LCFNEW|/Tmw2k/Unix/data/tmw2k_default.config|
4|0x403e0d8f|54214|0x000008|True|1|14 Oct 2005 19:50:26|14 Oct 2005
19:50:26|../../bin/aix4-r1/mrt|/bin/aix4-r1/mrt/atctl|
5|0x403e0f60|140773|0x000008|True|1|14 Oct 2005 19:50:26|14 Oct 2005
19:50:26|LCFNEW/Tmw2k/Unix/bin|/Tmw2k/Unix/bin/aix4-r1/gzip|
6|0x403e0f62|2049819|00000000|False|1|14 Oct 2005 19:50:27|14 Oct 2005
19:50:27|/opt/Tivoli/lcf_fab6/dat/8/cache|/lib/aix4-r1/libg++272.a|
7|0x403e0f63|2076642|00000000|False|1|14 Oct 2005 19:50:28|14 Oct 2005
19:50:28|/opt/Tivoli/lcf_fab6/dat/8/cache|/lib/aix4-r1/libstdc++272.a|
8|0x43207f80|55497|00000000|False|1|14 Oct 2005 19:50:29|14 Oct 2005
19:50:29|/opt/Tivoli/lcf_fab6/dat/8/cache|/lib/aix4-r1/libatrc.a|
9|0x403e0d8e|5850|00000000|False|1|14 Oct 2005 19:50:29|14 Oct 2005
19:50:29|/opt/Tivoli/lcf_fab6/dat/8/cache|/lib/aix4-r1/libatrcj.a|
10|0x4335ddc6|8131|00000000|True|1|14 Oct 2005 19:50:29|14 Oct 2005
19:50:29|LCFNEW|/Tmw2k/Unix/bin/getProfileInfo.sh|
11|0x43207f81|95810|00000000|False|1|14 Oct 2005 19:50:30|14 Oct 2005
19:50:30|/opt/Tivoli/lcf_fab6/dat/8/cache|/lib/aix4-r1/libccms272_lcf.a|
```

6.8 Agents files location

This section summarizes the location of the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint files:

- Integration Agent binaries

  $LCF_BINDIR/../TME/KTM/ktmcma$(EXE_EXTENSION)
- Integration Agent libraries
  - UNIX: $LCFROOT/lib/$INTERP
  - Windows: $LCFROOT/bin/$INTERP/mrt
- Log files:

  $LCF_DATDIR/LCFNEW/KTM/logs
- Metadata files

  $LCF_DATDIR/LCFNEW/KTM/ATTRLIB

  $LCF_DATDIR/LCFNEW/KTM/metadata
- Environment file

  $LCF_DATDIR/LCFNEW/KTM/KTMENV

6.9 Uninstall of the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint

This section describes the steps to remove the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint from your endpoint and the code from the Tivoli management region server.

6.9.1 Endpoint

A complete uninstall of the integration agent requires some steps at both Endpoint and Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint side.

- IBM Tivoli Monitoring 5.x environment

  a. Configure DataSeeding = ITM5.

  b. Stop the IBM Tivoli Monitoring engine.

c. Remove integration agent files from the agent using the following command:

```
witm61agt endpoint –r
```

Note: If you forget to stop the IBM Tivoli Monitoring engine, you will receive a lot of error messages because the remove operation will try to remove files that are currently in use.

- ITM 6.1 environment

a. Remove managed systems for this agent from the “Managed Systems Status workspace view”. Refer to 6.11.6, “Remove the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint from the TEP Workspace” on page 365.

6.9.2 Tivoli management region server

To uninstall the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint code from the TMR server, you can use the **wuninst** command like any other Tivoli product uninstallation (Example 6-3).

Example 6-3 ITM61AGT uninstallation

```
1) Run the below command to see the TAGs of the installed product:

[root@istanbul][/tmp/customrm]-> wuninst -list
Creating Log File (/tmp/wuninst.log)...
------------------------------------------------
    Uninstallable Products installed:
------------------------------------------------
ACF
DM_Advanced_Edition_TDS
ITM61AGT
ITMCmptSvcs
ITM_TEDW
InventoryGateway
InventoryServer
JCF411
JRE130
JRIM411
JavaHelp
Sentry2.0.2
TEC39_JRE
TEC_EVTHELP
TEC_JCONSOLE
TEC_SERVER
TEC_UI_SRVR
```

```
TMNT_3.6.2
mdist2gui
swdis
swdisgw
swdisjps
wuninst complete.
```

2) Run the below command to completely remove from your Tivoli Management Region environment the ITM61AGT code:

```
[root@istanbul][/tmp/customrm]-> wuninst ITM61AGT istanbul -rmfiles
```

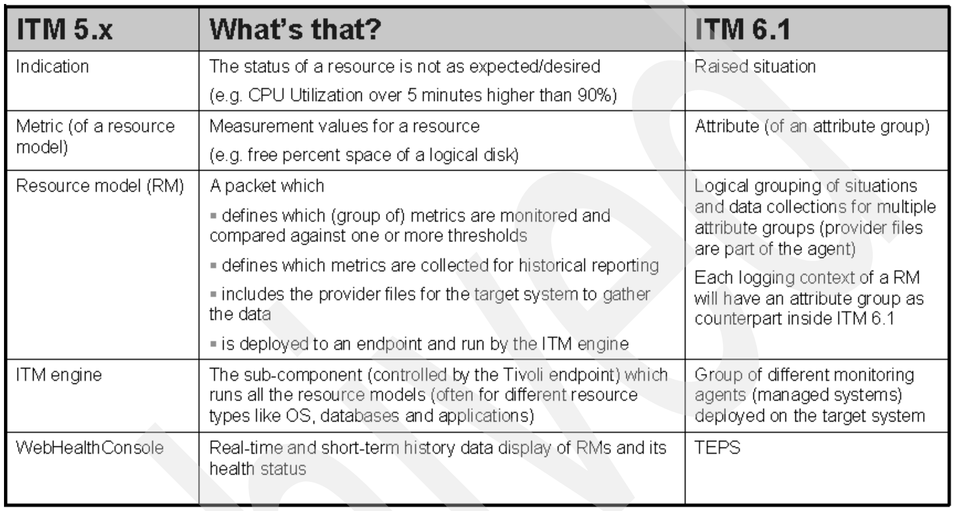
You will be prompted for a confirmation:

```
[root@istanbul][/tmp/customrm]-> wuninst ITM61AGT istanbul -rmfiles
Creating Log File (/tmp/wuninst.log)...
This command is about to remove ITM61AGT from the entire TMR.
Are you sure you want to continue? (y=yes, n=no) ?
```

Once the deinstall process completes, you are asked to run a **wchkdb -u** command to cleanup any entry left in the Tivoli Management Region server database.

6.10 Comparison between IBM Tivoli Monitoring 5.x and V6.1 terminology

Figure 6-7 analyzes the IBM Tivoli Monitoring 5.x terminology and translate them to the corresponding IBM Tivoli Monitoring 6.1.



ITM 5.x	What's that?	ITM 6.1
Indication	The status of a resource is not as expected/desired (e.g. CPU Utilization over 5 minutes higher than 90%)	Raised situation
Metric (of a resource model)	Measurement values for a resource (e.g. free percent space of a logical disk)	Attribute (of an attribute group)
Resource model (RM)	A packet which ▪ defines which (group of) metrics are monitored and compared against one or more thresholds ▪ defines which metrics are collected for historical reporting ▪ includes the provider files for the target system to gather the data ▪ is deployed to an endpoint and run by the ITM engine	Logical grouping of situations and data collections for multiple attribute groups (provider files are part of the agent) Each logging context of a RM will have an attribute group as counterpart inside ITM 6.1
ITM engine	The sub-component (controlled by the Tivoli endpoint) which runs all the resource models (often for different resource types like OS, databases and applications)	Group of different monitoring agents (managed systems) deployed on the target system
WebHealthConsole	Real-time and short-term history data display of RMs and its health status	TEPS

Figure 6-7 Differences between IBM Tivoli Monitoring 5.x and V6.1 terminology

6.11 Workspaces, views, situations, and queries

A workspace is the working area of the Tivoli Enterprise Portal application window. At the left of the workspace is a Navigator that you use to select the workspace you want to see.

As you select items in the Navigator, the workspace presents views pertinent to your selection. Each workspace has at least one view. Some views contain links to workspaces. If an item has multiple workspaces created for it, you can right-click the item, select Workspaces, and click another workspace to open. Every workspace has a set of properties associated with it.

This Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint provides predefined workspaces for the resource models shipped by IBM. You cannot modify or delete the predefined workspaces, but you can create new workspaces by editing them and saving the changes with a different name.

For more information about creating, customizing, and working with workspaces, see *IBM Tivoli Monitoring, Version 6.1.0, User's Guide*, SC32-9409.

For a list of the predefined workspaces for this monitoring agent and a description of each workspace, refer to the Predefined workspaces section and the information in that section for each individual workspace that can be found in *IBM Tivoli Monitoring: IBM Tivoli Monitoring 5.x Endpoint Agent User's Guide*, SC32-9490.

6.11.1 Health Navigator item

The Health workspaces are applicable to all of the IBM Tivoli Monitoring resource models.

Table 6-4 contains a list of predefined workspaces for the Health Navigator item. These workspaces are listed with the Navigator items from which they are accessed.

Table 6-4 Predefined Health workspaces by Navigator item

Navigator item	Workspaces
Logged Data	Logged Data
Messages	Messages
Resource Model	Resource Model Overview Indication Instances Indication Metrics

The sections that follow contain descriptions of each of the predefined workspaces in Table 6-4.

The workspaces are organized by the Navigator item to which the workspaces are relevant.

Logged Data workspaces

Use this workspace to view the following information about logged data that does not match one of the definitions included for the understood logging statements (for example, custom resource models):

- Time the data was collected
- Category
- Internal name of the resource model
- Context
- Resource
- Instance

- Name of the IBM Tivoli Monitoring metric
- Type of metric (numeric or string)
- String value of the metric
- Integer value of the metric
- Float value of the metric

This workspace has one table view containing this information, KTM_LOGGED_DATA.

Messages workspaces

Use this workspace to view information about messages issued by the agent to represent the state of the operation of the agent. This workspace has one view: Messages table. The following information is provided in this table:

- Time the data was collected
- Type of message such as information or error
- Text that describes the condition

6.11.2 Resource Model Navigator item

Resource Model Overview workspace

Use this workspace to view information about the health of your resource models.

This resource model has two views:

- Resource Model Health bar chart that shows all of the resource models running on this endpoint that have health of less than 100 percent
- Resource Model Overview table, which contains the following information:
  - Time the data was collected
  - Name of the profile used to distribute the instance of the resource model
  - Internal name of the resource model
  - Operational state of the resource model
  - Cycle time or the resource model
  - Resource model health value of 0 – 100

    A twistie (rightward-pointing or downward-pointing triangle icon) is active for each resource model where the health is less than 100 and the status is running or delayed. You can click the twistie to see more detailed data:

    - The Resource Model Overview workspace contains links to the Indications Instances workspace. These links are the twisties for the resource models in the Resource Model Overview table view. Click on a twistie to see the Indications Instances workspace.

- The Indications Instances workspace contains links to the Indications Metrics workspace. These links are the twisties for the indication instances in the Indication Instances table view. Click on a twistie to see the Indication Metrics table view.

Indications Instances workspace

Use this workspace to view information about the instances of the indications.

This workspace has three views:

- Resource Model table
- Health Trend plot chart
- Indication Instances table

The Indication Instances table contains links to the indication metrics.

Indication Metrics workspace

Use this workspace to view data about the resource model indications.

This workspace has three views:

- Resource Model table
- Indication Instance table
- Indication Metrics table

6.11.3 Predefined workspaces

"Appendix B. Workspaces reference" of *IBM Tivoli Monitoring: IBM Tivoli Monitoring 5.x Endpoint Agent User's Guide*, SC32-9490, contains information about all of the predefined workspaces provided with the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint. These workspaces are organized by the following resource model categories in the Navigator tree under the IBM Tivoli Monitoring 5.x Endpoint Agent node:

- Health Navigator item
- DB2 Navigator item
- Oracle Navigator item
- Solaris Navigator item
- UNIX-Linux Navigator item
- WebSphere Application Server Navigator item
- Windows Navigator item
- Other IBM Tivoli Monitoring component software workspaces

6.11.4 Situations and queries

As with any regular IBM Tivoli Monitoring 6.1 agent, the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint supports customer-defined situations and queries.

They can be distributed either to all endpoints using the IBM Tivoli Monitoring 6.1 agent managed system list or only to selected subnodes (endpoint PAC level).

Some situations are provided as an example and to make seeding behave the same way in both Windows and UNIX/Linux installers.

Situations are disabled by default but associated with the correct report nodes in the tree.

Supplied situations are:

- KTM_Missing_Metadata

  Active when data logged to the generic table is found. Can be used to show metadata problems or missing metadata for customized resource models.
- KTM_Health_Indication

  Shows running or delaying resource models with a health of less than 100.
- KTM_Resource_Model_Status

  Shows resource models in non-functional states (not equal to running, disabled, stopped, scheduled or delaying) for at least two consecutive samples.

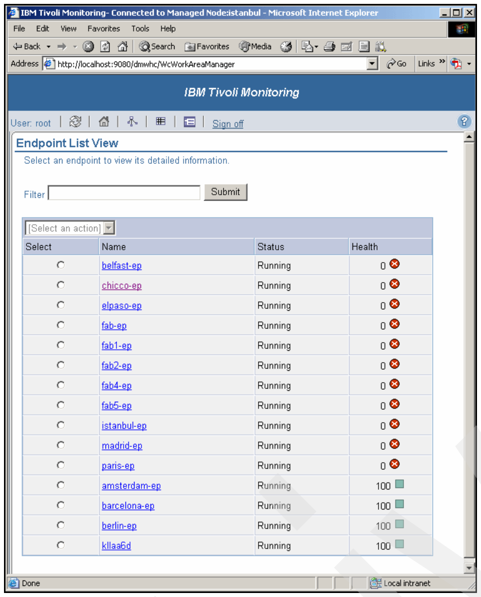
You can refer to Chapter 8, “Real-life scenarios” on page 469 for more information about situations and queries.

6.11.5 Web Health Console versus Tivoli Enterprise Portal

The data that is available in the Web Health Console is also available in the Tivoli Enterprise Portal. The organization is similar, but the specific views and their contents are different because of differences in how information is displayed in the two user interfaces. The Web Health Console has two different top level views: Endpoint view and Resource Model view. The Tivoli Enterprise Portal displays data much like the Web Health Console Endpoint View.

This section provides some comparisons between the Web Health Console and the Tivoli Enterprise Portal.

Web Health Console Endpoint List View


IBM Tivoli Monitoring- Connected to Managed Node:istanbul - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost:9080/dmwhc/WcWorkAreaManager
Go Links
IBM Tivoli Monitoring
User: root
Sign off
Done
Local intranet

Endpoint List View

Select an endpoint to view its detailed information.

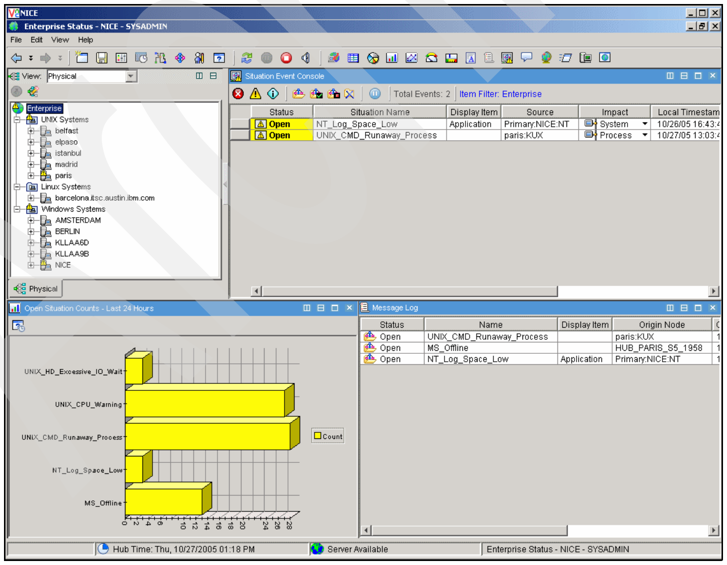
Filter [ ] Submit

[Select an action]

Select	Name	Status	Health
○	belfast-ep	Running	0 ⊗
○	chicco-ep	Running	0 ⊗
○	elpaso-ep	Running	0 ⊗
○	fab-ep	Running	0 ⊗
○	fab1-ep	Running	0 ⊗
○	fab2-ep	Running	0 ⊗
○	fab4-ep	Running	0 ⊗
○	fab5-ep	Running	0 ⊗
○	istanbul-ep	Running	0 ⊗
○	madrid-ep	Running	0 ⊗
○	paris-ep	Running	0 ⊗
○	amsterdam-ep	Running	100 🟩
○	barcelona-ep	Running	100 🟩
○	berlin-ep	Running	100 🟩
○	kllaa6d	Running	100 🟩

Figure 6-8 Web Health Console Endpoint List View

Tivoli Enterprise Portal Agents View

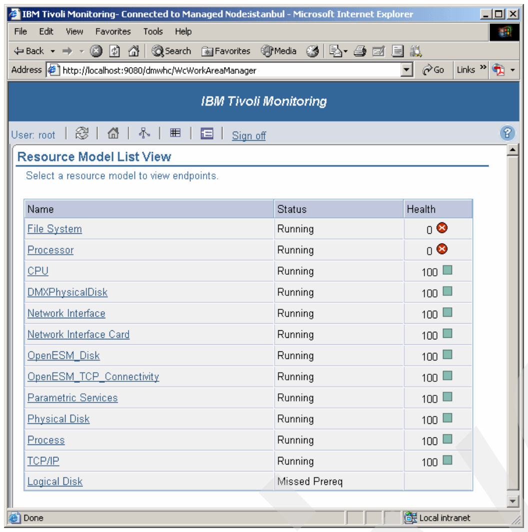


Status	Situation Name	Display Item	Source	Impact	Local Timestam
Open	NT_Log_Space_Low	Application	Primary:NICE:NT	System	10/26/05 16:43:4
Open	UNIX_CMD_Runaway_Process		paris:KUX	Process	10/27/05 13:03:4

Status	Name	Display Item	Origin Node
Open	UNIX_CMD_Runaway_Process		paris:KUX
Open	MS_Offline		HUB_PARIS_S5_1958
Open	NT_Log_Space_Low	Application	Primary:NICE:NT

Figure 6-9 Tivoli Enterprise Portal Agents View

Web Health Console Resource Model List View



IBM Tivoli Monitoring- Connected to Managed Node:istanbul - Microsoft Internet Explorer

Address http://localhost:9080/dmwhc/WcWorkAreaManager

IBM Tivoli Monitoring

User: root
Sign off

Resource Model List View

Select a resource model to view endpoints.

Name	Status	Health
File System	Running	0 ⊗
Processor	Running	0 ⊗
CPU	Running	100 🟩
DMXPhysicalDisk	Running	100 🟩
Network Interface	Running	100 🟩
Network Interface Card	Running	100 🟩
OpenESM_Disk	Running	100 🟩
OpenESM_TCP_Connectivity	Running	100 🟩
Parametric Services	Running	100 🟩
Physical Disk	Running	100 🟩
Process	Running	100 🟩
TCP/IP	Running	100 🟩
Logical Disk	Missed Prereq	

Done | Local intranet

Figure 6-10   Web Health Console Resource Model List View

Tivoli Enterprise Portal Resource Model Data View

The Tivoli Enterprise Portal does not have a resource model view corresponding to the Web Health Console, but you can select a single resource model from the Agent menu and obtain a view similar to the Figure 6-11.

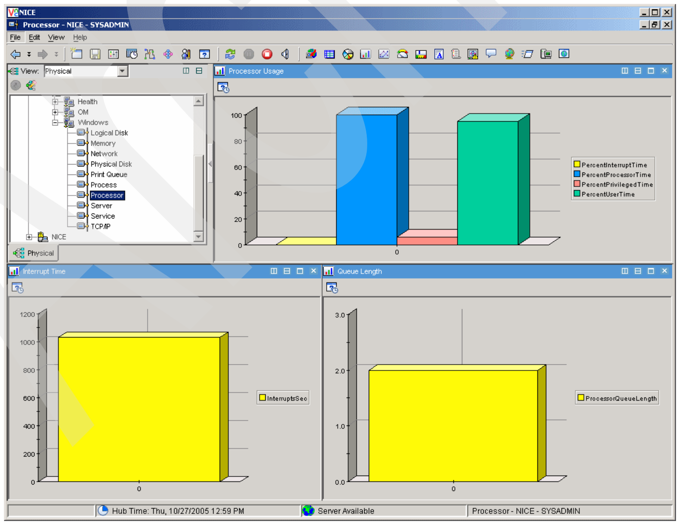
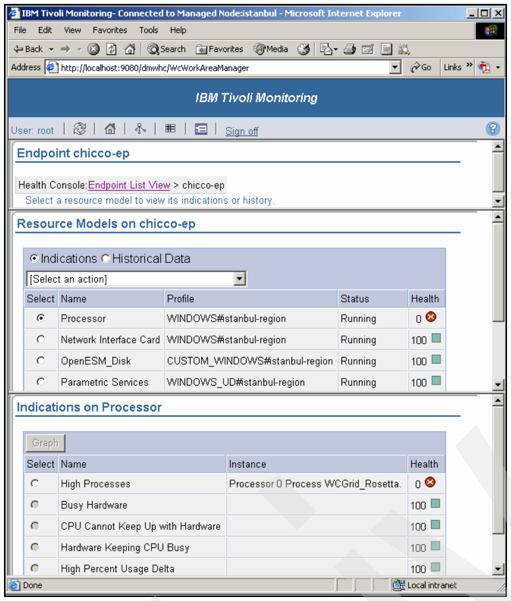


Figure 6-11   Tivoli Enterprise Portal Resource Model Data View

Web Health Console Resource Model Health View



IBM Tivoli Monitoring- Connected to Managed Node:istanbul - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:9080/dmwhc/WcWorkAreaManager
IBM Tivoli Monitoring
User: root
Sign off

Endpoint chicco-ep

Health Console: Endpoint List View > chicco-ep

Select a resource model to view its indications or history.

Resource Models on chicco-ep

(● Indications ○ Historical Data)

[Select an action]

Select	Name	Profile	Status	Health
●	Processor	WINDOWS#istanbul-region	Running	0 ⊗
○	Network Interface Card	WINDOWS#istanbul-region	Running	100 🟩
○	OpenESM_Disk	CUSTOM_WINDOWS#istanbul-region	Running	100 🟩
○	Parametric Services	WINDOWS_UD#istanbul-region	Running	100 🟩

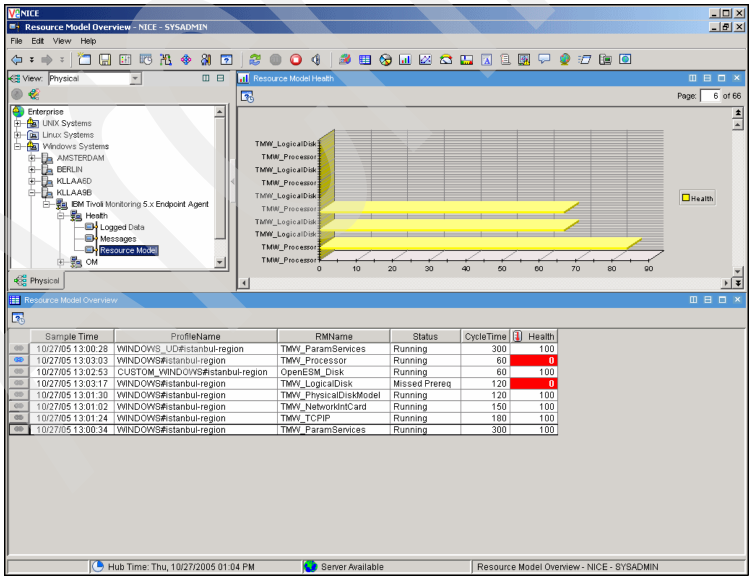
Indications on Processor

Graph

Select	Name	Instance	Health
○	High Processes	Processor 0 Process WCGrid_Rosetta.	0 ⊗
○	Busy Hardware		100 🟩
○	CPU Cannot Keep Up with Hardware		100 🟩
○	Hardware Keeping CPU Busy		100 🟩
○	High Percent Usage Delta		100 🟩

Figure 6-12 Web Health Console Resource Model Health View

Tivoli Enterprise Portal Resource Model Health View



Sample Time	ProfileName	RMName	Status	CycleTime	Health
10/27/05 13:00:28	WINDOWS_UD#istanbul-region	TMW_ParamServices	Running	300	100
10/27/05 13:03:03	WINDOWS#istanbul-region	TMW_Processor	Running	60	0
10/27/05 13:02:53	CUSTOM_WINDOWS#istanbul-region	OpenESM_Disk	Running	60	100
10/27/05 13:03:17	WINDOWS#istanbul-region	TMW_LogicalDisk	Missed Prereq	120	0
10/27/05 13:01:30	WINDOWS#istanbul-region	TMW_PhysicalDiskModel	Running	120	100
10/27/05 13:01:02	WINDOWS#istanbul-region	TMW_NetworkIntCard	Running	150	100
10/27/05 13:01:24	WINDOWS#istanbul-region	TMW_TCPIP	Running	180	100
10/27/05 13:00:34	WINDOWS#istanbul-region	TMW_ParamServices	Running	300	100

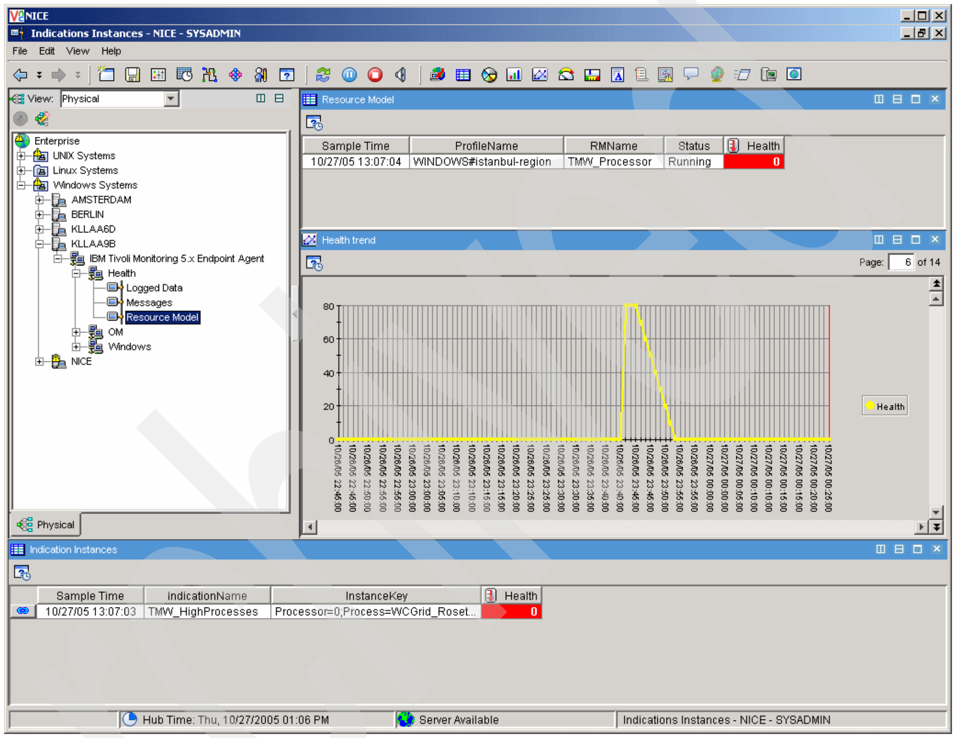
Hub Time: Thu, 10/27/2005 01:04 PM
Server Available
Resource Model Overview - NICE - SYSADMIN

Figure 6-13 Tivoli Enterprise Portal Resource Model Health View

Tivoli Enterprise Portal Resource Model Health trend

The Tivoli Enterprise Portal provides you the ability to analyze the Resource Model Health trend. Figure 6-14 shows an example.

This possibility enables you to know exactly when a problem started to occur on a specific system, so that you can make an accurate problem determination trying to isolate and understand the reason for the failure.



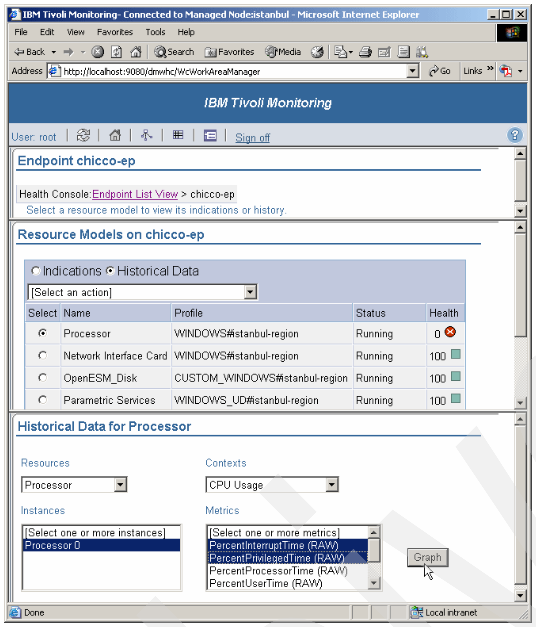
Sample Time	ProfileName	RMName	Status	Health
10/27/05 13:07:04	WINDOWS#istanbul-region	TMW_Processor	Running	0

Sample Time	IndicationName	InstanceKey	Health
10/27/05 13:07:03	TMW_HighProcesses	Processor=0;Process=WCGrid_Roset...	0

Hub Time: Thu, 10/27/2005 01:06 PM | Server Available | Indications Instances - NICE - SYSADMIN

Figure 6-14 Tivoli Enterprise Portal Resource Model Health Trend

Web Health Console Metrics selection to create a graph


IBM Tivoli Monitoring- Connected to Managed Node:istanbul - Microsoft Internet Explorer

Address: http://localhost:9080/dmwhc/WcWorkAreaManager

IBM Tivoli Monitoring

User: root | Sign off

Endpoint chicco-ep

Health Console: Endpoint List View > chicco-ep

Select a resource model to view its indications or history.

Resource Models on chicco-ep

Indications [✓ Historical Data]

[Select an action]

Select	Name	Profile	Status	Health
●	Processor	WINDOWS#stanbul-region	Running	0
○	Network Interface Card	WINDOWS#stanbul-region	Running	100
○	OpenESM_Disk	CUSTOM_WINDOWS#stanbul-region	Running	100
○	Parametric Services	WINDOWS_UD#stanbul-region	Running	100

Historical Data for Processor

Resources: Processor

Contexts: CPU Usage

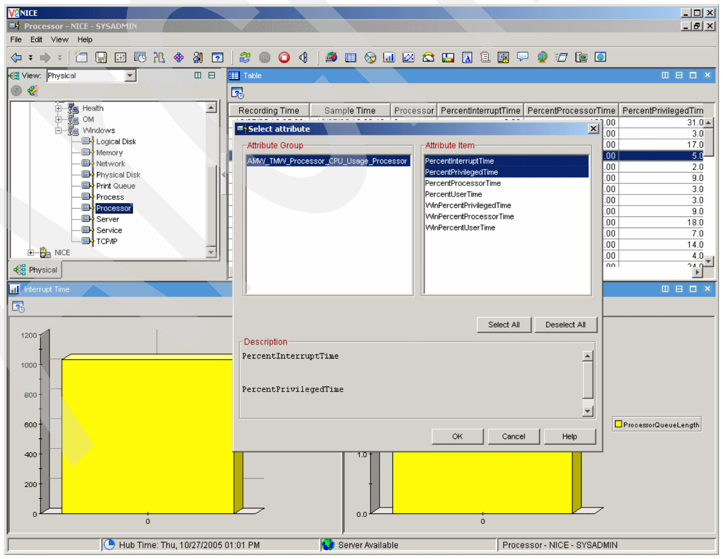
Instances:
- [Select one or more instances]
- Processor 0

Metrics:
- [Select one or more metrics]
- PercentInterruptTime (RAW)
- PercentPrivilegedTime (RAW)
- PercentProcessorTime (RAW)
- PercentUserTime (RAW)

Graph

Figure 6-15   Web Health Console Metrics selection

Tivoli Enterprise Portal Attributes selection to create a graph


NICE

Processor - NICE - SYSADMIN

View: Physical

- Health
- OM
- Windows
  - Logical Disk
  - Memory
  - Network
  - Physical Disk
  - Print Queue
  - Process
  - Processor
  - Server
  - Service
  - TCP/IP
- NICE

Table: Recording Time | Sample Time | Processor | PercentInterruptTime | PercentProcessorTime | PercentPrivilegedTim

Select attribute

Attribute Group:
- AMW_TMW_Processor_CPU_Usage_Processor

Attribute Item:
- PercentInterruptTime
- PercentPrivilegedTime
- PercentProcessorTime
- PercentUserTime
- WinPercentPrivilegedTime
- WinPercentProcessorTime
- WinPercentUserTime

Select All | Deselect All

Description:

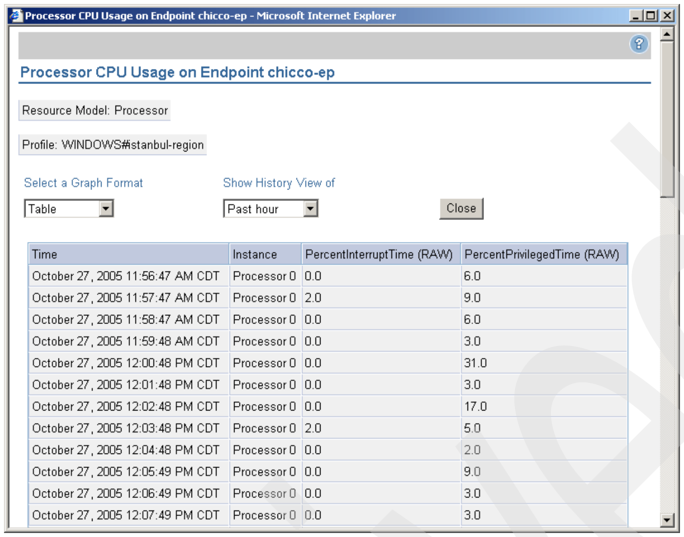
```
PercentInterruptTime

PercentPrivilegedTime
```

OK | Cancel | Help

Hub Time: Thu, 10/27/2005 01:01 PM | Server Available | Processor - NICE - SYSADMIN

Figure 6-16   Tivoli Enterprise Portal Attributes selection

Web Health Console Table chart


Processor CPU Usage on Endpoint chicco-ep - Microsoft Internet Explorer

Processor CPU Usage on Endpoint chicco-ep

Resource Model: Processor

Profile: WINDOWS#istanbul-region

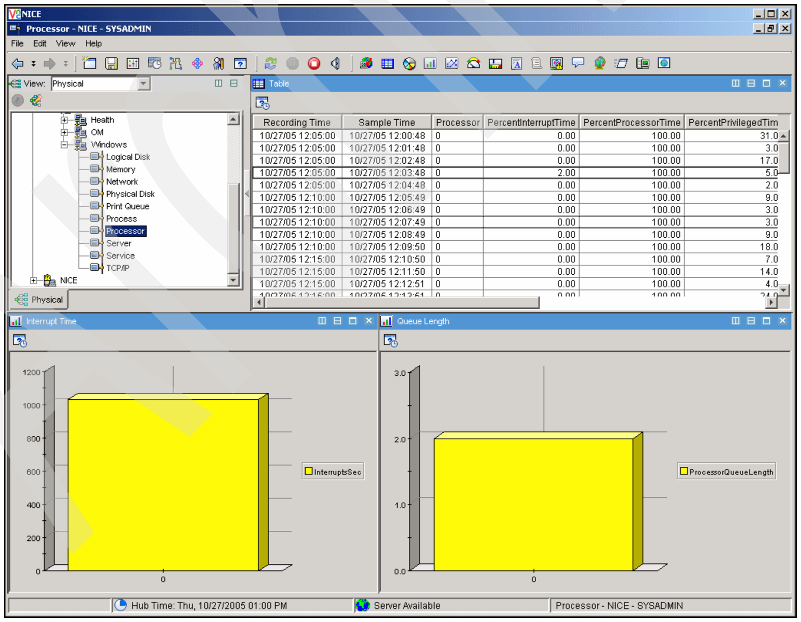
Select a Graph Format: Table

Show History View of: Past hour

Time	Instance	PercentInterruptTime (RAW)	PercentPrivilegedTime (RAW)
October 27, 2005 11:56:47 AM CDT	Processor 0	0.0	6.0
October 27, 2005 11:57:47 AM CDT	Processor 0	2.0	9.0
October 27, 2005 11:58:47 AM CDT	Processor 0	0.0	6.0
October 27, 2005 11:59:48 AM CDT	Processor 0	0.0	3.0
October 27, 2005 12:00:48 PM CDT	Processor 0	0.0	31.0
October 27, 2005 12:01:48 PM CDT	Processor 0	0.0	3.0
October 27, 2005 12:02:48 PM CDT	Processor 0	0.0	17.0
October 27, 2005 12:03:48 PM CDT	Processor 0	2.0	5.0
October 27, 2005 12:04:48 PM CDT	Processor 0	0.0	2.0
October 27, 2005 12:05:49 PM CDT	Processor 0	0.0	9.0
October 27, 2005 12:06:49 PM CDT	Processor 0	0.0	3.0
October 27, 2005 12:07:49 PM CDT	Processor 0	0.0	3.0

Figure 6-17 Web Health Console Table chart

Tivoli Enterprise Portal Table chart

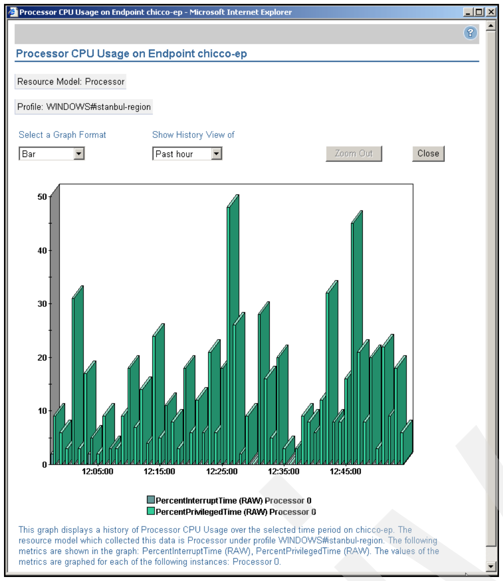


Recording Time	Sample Time	Processor	PercentInterruptTime	PercentProcessorTime	PercentPrivilegedTim
10/27/05 12:05:00	10/27/05 12:00:48	0	0.00	100.00	31.0
10/27/05 12:05:00	10/27/05 12:01:48	0	0.00	100.00	3.0
10/27/05 12:05:00	10/27/05 12:02:48	0	0.00	100.00	17.0
10/27/05 12:05:00	10/27/05 12:03:48	0	2.00	100.00	5.0
10/27/05 12:05:00	10/27/05 12:04:48	0	0.00	100.00	2.0
10/27/05 12:10:00	10/27/05 12:05:49	0	0.00	100.00	9.0
10/27/05 12:10:00	10/27/05 12:06:49	0	0.00	100.00	3.0
10/27/05 12:10:00	10/27/05 12:07:49	0	0.00	100.00	3.0
10/27/05 12:10:00	10/27/05 12:08:49	0	0.00	100.00	9.0
10/27/05 12:10:00	10/27/05 12:09:50	0	0.00	100.00	18.0
10/27/05 12:15:00	10/27/05 12:10:50	0	0.00	100.00	7.0
10/27/05 12:15:00	10/27/05 12:11:50	0	0.00	100.00	14.0
10/27/05 12:15:00	10/27/05 12:12:51	0	0.00	100.00	4.0

Hub Time: Thu, 10/27/2005 01:00 PM | Server Available | Processor - NICE - SYSADMIN

Figure 6-18 Tivoli Enterprise Portal Table chart

Web Health Console Bar chart



Processor CPU Usage on Endpoint chicco-ep - Microsoft Internet Explorer
Processor CPU Usage on Endpoint chicco-ep
Select a Graph Format
Show History View of
Zoom Out
Close
50
40
30
20
10
0
12:05:00
12:15:00
12:25:00
12:35:00
12:45:00

Resource Model: Processor

Profile: WINDOWS#istanbul-region

Select a Graph Format: Bar

Show History View of: Past hour

▨ PercentInterruptTime (RAW) Processor 0
🟩 PercentPrivilegedTime (RAW) Processor 0

This graph displays a history of Processor CPU Usage over the selected time period on chicco-ep. The resource model which collected this data is Processor under profile WINDOWS#istanbul-region. The following metrics are shown in the graph: PercentInterruptTime (RAW), PercentPrivilegedTime (RAW). The values of the metrics are graphed for each of the following instances: Processor 0.

Figure 6-19   Web Health Console Bar chart

Tivoli Enterprise Portal Bar chart

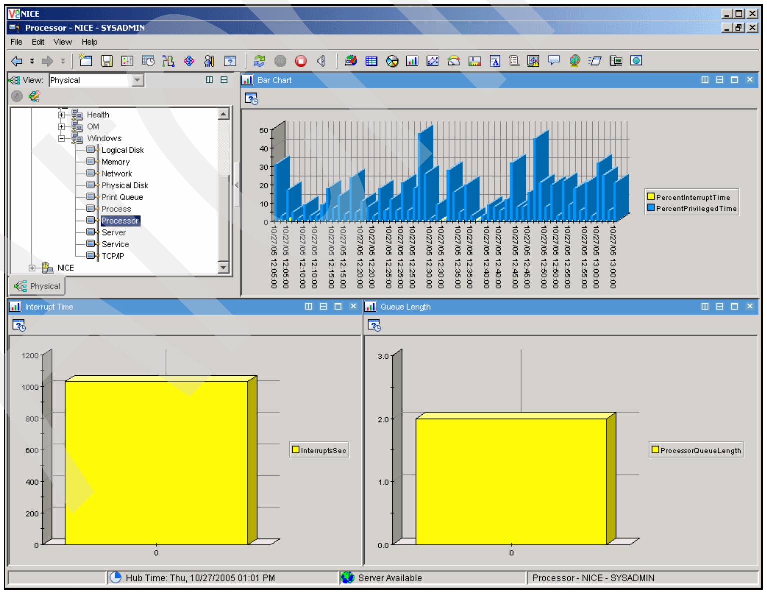
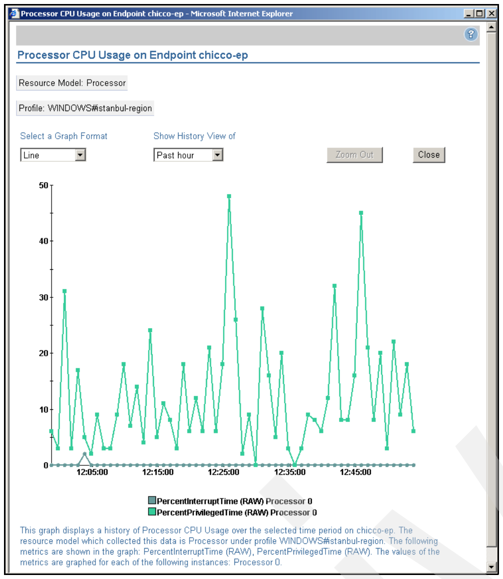


Figure 6-20   Tivoli Enterprise portal bar chart

Web Health Console Line chart



Processor CPU Usage on Endpoint chicco-ep - Microsoft Internet Explorer

Processor CPU Usage on Endpoint chicco-ep

Resource Model: Processor

Profile: WINDOWS#stanbul-region

Select a Graph Format: Line

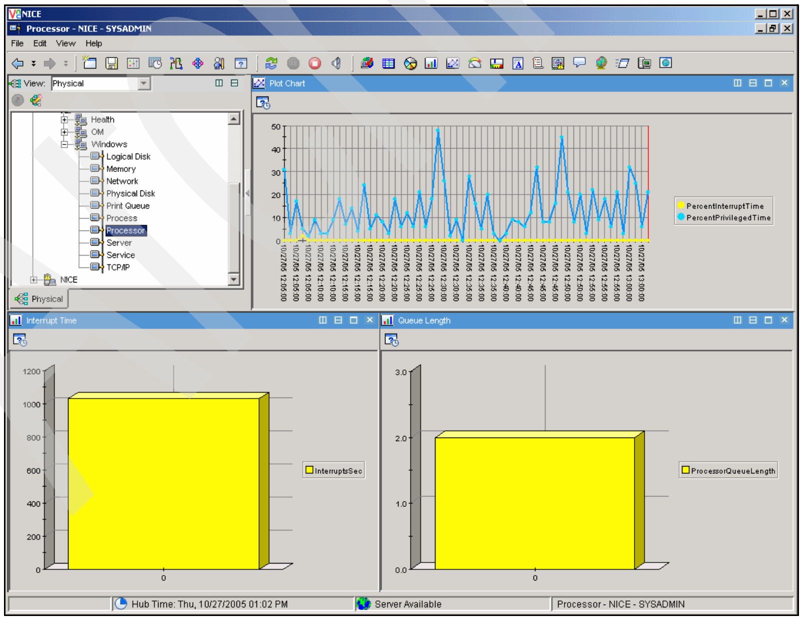
Show History View of: Past hour

🟩 PercentInterruptTime (RAW) Processor 0
🟩 PercentPrivilegedTime (RAW) Processor 0

This graph displays a history of Processor CPU Usage over the selected time period on chicco-ep. The resource model which collected this data is Processor under profile WINDOWS#stanbul-region. The following metrics are shown in the graph: PercentInterruptTime (RAW), PercentPrivilegedTime (RAW). The values of the metrics are graphed for each of the following instances: Processor 0.

Figure 6-21 Web Health Console Line chart

Tivoli Enterprise Portal Plot chart



Hub Time: Thu, 10/27/2005 01:02 PM | Server Available | Processor - NICE - SYSADMIN

Figure 6-22 Tivoli Enterprise Portal Plot chart

6.11.6 Remove the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint from the TEP Workspace

The Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint has three entries in the TEP Navigation Tree:

- TM

  This is the main entry in the tree for the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint, and it is listed as IBM Tivoli Monitoring 5.x Endpoint Agent.

  If you have multiple endpoints running on the same machine, a single TM entry will have the host name of the machine hosting the endpoints, and a subtree for each endpoint label and a KTM an AMX/AMW subtree for each.

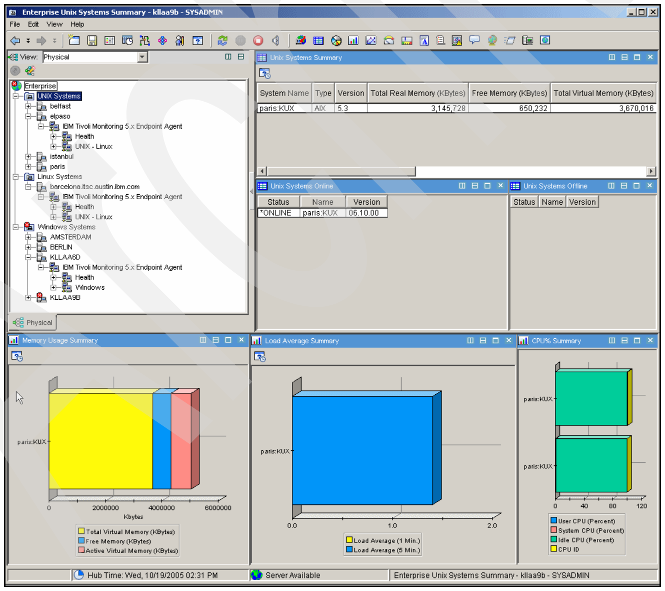
- KTM

  This entry represents the Health subtree.

- AMX/AMW

  This entry represents the logged data and change based on the OS type.

  What you see in the tree will be UNIX - Linux for UNIX and Linux Endpoints, Windows for Windows Endpoints. Figure 6-23 shows an example.



```
Enterprise
  UNIX Systems
    belfast
    elpaso
      IBM Tivoli Monitoring 5.x Endpoint Agent
        Health
        UNIX - Linux
    istanbul
    paris
  Linux Systems
    barcelona.itsc.austin.ibm.com
      IBM Tivoli Monitoring 5.x Endpoint Agent
        Health
        UNIX - Linux
  Windows Systems
    AMSTERDAM
    BERLIN
    KLLAA6D
      IBM Tivoli Monitoring 5.x Endpoint Agent
        Health
        Windows
    KLLAA9B
```

Unix Systems Summary

System Name	Type	Version	Total Real Memory (KBytes)	Free Memory (KBytes)	Total Virtual Memory (KBytes)
paris:KUX	AIX	5.3	3,145,728	650,232	3,670,016

Unix Systems Online

Status	Name	Version
*ONLINE	paris:KUX	06.10.00

Unix Systems Offline

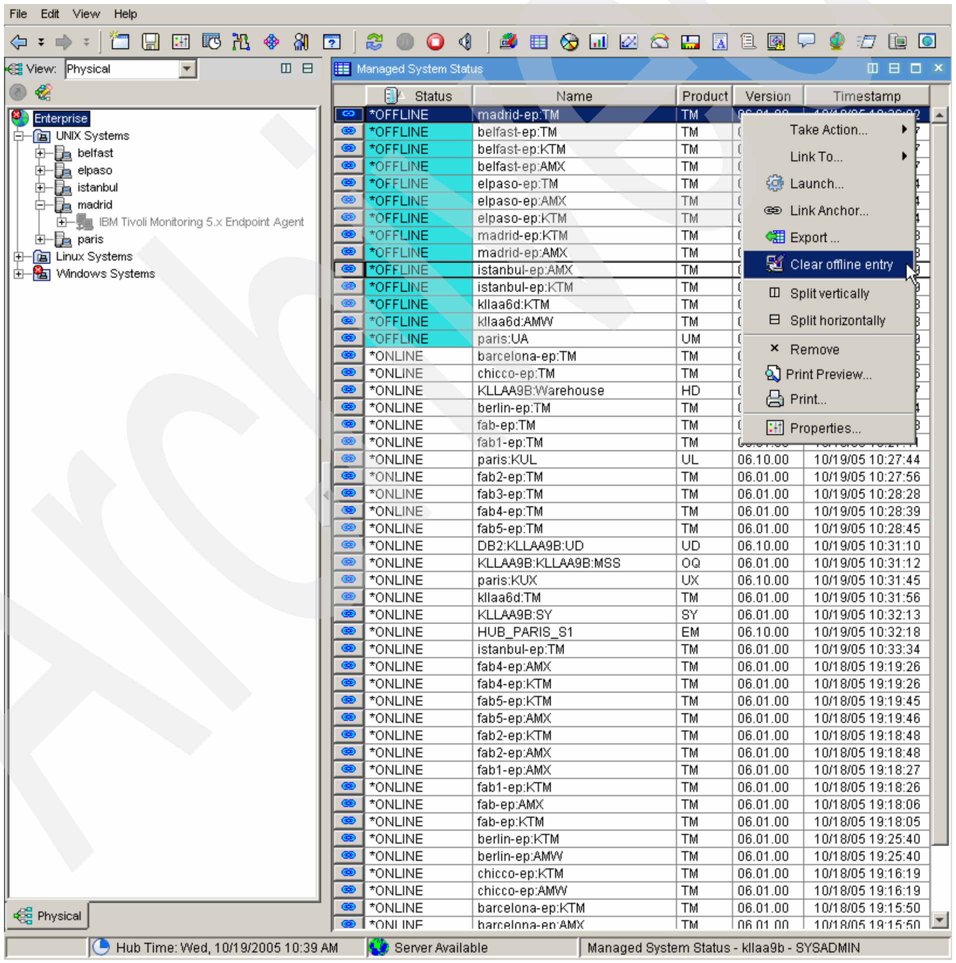
Status	Name	Version

Figure 6-23 Example of UNIX, Linux, and Windows trees

If you want to remove one or more OFFLINE entries, you should first open the Managed System Status workspace view. From the TEP console, perform these steps:

1. Select **Enterprise.**
2. Right-click to get the list of available workspaces view and select **Managed System Status.**
3. Sort the view (optional).
4. Select the OFFLINE item you want to remove, right-click to get the available options, and select **Clear offline entry.**

   Figure 6-24 shows these steps.

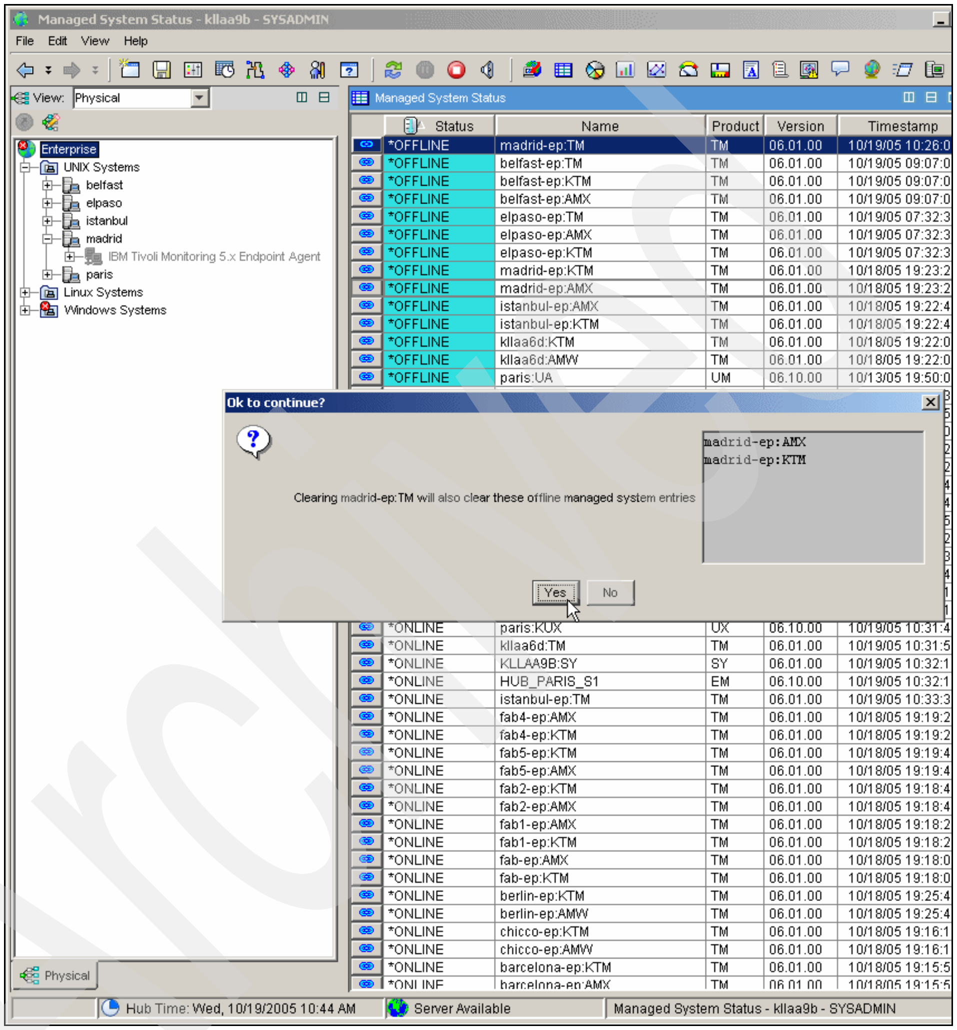


Status	Name	Product	Version	Timestamp
*OFFLINE	madrid-ep:TM	TM	[illegible]	[illegible]
*OFFLINE	belfast-ep:TM	TM	[illegible]	[illegible]
*OFFLINE	belfast-ep:KTM	TM	[illegible]	[illegible]
*OFFLINE	belfast-ep:AMX	TM	[illegible]	[illegible]
*OFFLINE	elpaso-ep:TM	TM	[illegible]	[illegible]
*OFFLINE	elpaso-ep:AMX	TM	[illegible]	[illegible]
*OFFLINE	elpaso-ep:KTM	TM	[illegible]	[illegible]
*OFFLINE	madrid-ep:KTM	TM	[illegible]	[illegible]
*OFFLINE	madrid-ep:AMX	TM	[illegible]	[illegible]
*OFFLINE	istanbul-ep:AMX	TM	[illegible]	[illegible]
*OFFLINE	istanbul-ep:KTM	TM	[illegible]	[illegible]
*OFFLINE	kllaa6d:KTM	TM	[illegible]	[illegible]
*OFFLINE	kllaa6d:AMW	TM	[illegible]	[illegible]
*OFFLINE	paris:UA	UM	[illegible]	[illegible]
*ONLINE	barcelona-ep:TM	TM	[illegible]	[illegible]
*ONLINE	chicco-ep:TM	TM	[illegible]	[illegible]
*ONLINE	KLLAA9B:Warehouse	HD	[illegible]	[illegible]
*ONLINE	berlin-ep:TM	TM	[illegible]	[illegible]
*ONLINE	fab-ep:TM	TM	[illegible]	[illegible]
*ONLINE	fab1-ep:TM	TM	[illegible]	[illegible]
*ONLINE	paris:KUL	UL	06.10.00	10/19/05 10:27:44
*ONLINE	fab2-ep:TM	TM	06.01.00	10/19/05 10:27:56
*ONLINE	fab3-ep:TM	TM	06.01.00	10/19/05 10:28:28
*ONLINE	fab4-ep:TM	TM	06.01.00	10/19/05 10:28:39
*ONLINE	fab5-ep:TM	TM	06.01.00	10/19/05 10:28:45
*ONLINE	DB2:KLLAA9B:UD	UD	06.10.00	10/19/05 10:31:10
*ONLINE	KLLAA9B:KLLAA9B:MSS	OQ	06.01.00	10/19/05 10:31:12
*ONLINE	paris:KUX	UX	06.10.00	10/19/05 10:31:45
*ONLINE	kllaa6d:TM	TM	06.01.00	10/19/05 10:31:56
*ONLINE	KLLAA9B:SY	SY	06.01.00	10/19/05 10:32:13
*ONLINE	HUB_PARIS_S1	EM	06.10.00	10/19/05 10:32:18
*ONLINE	istanbul-ep:TM	TM	06.01.00	10/19/05 10:33:34
*ONLINE	fab4-ep:AMX	TM	06.01.00	10/18/05 19:19:26
*ONLINE	fab4-ep:KTM	TM	06.01.00	10/18/05 19:19:26
*ONLINE	fab5-ep:KTM	TM	06.01.00	10/18/05 19:19:45
*ONLINE	fab5-ep:AMX	TM	06.01.00	10/18/05 19:19:46
*ONLINE	fab2-ep:KTM	TM	06.01.00	10/18/05 19:18:48
*ONLINE	fab2-ep:AMX	TM	06.01.00	10/18/05 19:18:48
*ONLINE	fab1-ep:AMX	TM	06.01.00	10/18/05 19:18:27
*ONLINE	fab1-ep:KTM	TM	06.01.00	10/18/05 19:18:26
*ONLINE	fab-ep:AMX	TM	06.01.00	10/18/05 19:18:06
*ONLINE	fab-ep:KTM	TM	06.01.00	10/18/05 19:18:05
*ONLINE	berlin-ep:KTM	TM	06.01.00	10/18/05 19:25:40
*ONLINE	berlin-ep:AMW	TM	06.01.00	10/18/05 19:25:40
*ONLINE	chicco-ep:KTM	TM	06.01.00	10/18/05 19:16:19
*ONLINE	chicco-ep:AMW	TM	06.01.00	10/18/05 19:16:19
*ONLINE	barcelona-ep:KTM	TM	06.01.00	10/18/05 19:15:50
*ONLINE	barcelona-ep:AMX	TM	06.01.00	10/18/05 19:15:50

Hub Time: Wed, 10/19/2005 10:39 AM | Server Available | Managed System Status - kllaa9b - SYSADMIN

Figure 6-24 Clear offline entry menu item

5. To remove all entries about one Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint, you can just remove the TM entry that is the root of the tree and you will be prompted for the removal of any correlated object (Figure 6-25).


Managed System Status - kllaa9b - SYSADMIN

File Edit View Help

View: Physical

- Enterprise
  - UNIX Systems
    - belfast
    - elpaso
    - istanbul
    - madrid
      - IBM Tivoli Monitoring 5.x Endpoint Agent
    - paris
  - Linux Systems
  - Windows Systems

Managed System Status

Status	Name	Product	Version	Timestamp
*OFFLINE	madrid-ep:TM	TM	06.01.00	10/19/05 10:26:0
*OFFLINE	belfast-ep:TM	TM	06.01.00	10/19/05 09:07:0
*OFFLINE	belfast-ep:KTM	TM	06.01.00	10/19/05 09:07:0
*OFFLINE	belfast-ep:AMX	TM	06.01.00	10/19/05 09:07:0
*OFFLINE	elpaso-ep:TM	TM	06.01.00	10/19/05 07:32:3
*OFFLINE	elpaso-ep:AMX	TM	06.01.00	10/19/05 07:32:3
*OFFLINE	elpaso-ep:KTM	TM	06.01.00	10/19/05 07:32:3
*OFFLINE	madrid-ep:KTM	TM	06.01.00	10/18/05 19:23:2
*OFFLINE	madrid-ep:AMX	TM	06.01.00	10/18/05 19:23:2
*OFFLINE	istanbul-ep:AMX	TM	06.01.00	10/18/05 19:22:4
*OFFLINE	istanbul-ep:KTM	TM	06.01.00	10/18/05 19:22:4
*OFFLINE	kllaa6d:KTM	TM	06.01.00	10/18/05 19:22:0
*OFFLINE	kllaa6d:AMW	TM	06.01.00	10/18/05 19:22:0
*OFFLINE	paris:UA	UM	06.10.00	10/13/05 19:50:0
*ONLINE	paris:KUX	UX	06.10.00	10/19/05 10:31:4
*ONLINE	kllaa6d:TM	TM	06.01.00	10/19/05 10:31:5
*ONLINE	KLLAA9B:SY	SY	06.01.00	10/19/05 10:32:1
*ONLINE	HUB_PARIS_S1	EM	06.10.00	10/19/05 10:32:1
*ONLINE	istanbul-ep:TM	TM	06.01.00	10/19/05 10:33:3
*ONLINE	fab4-ep:AMX	TM	06.01.00	10/18/05 19:19:2
*ONLINE	fab4-ep:KTM	TM	06.01.00	10/18/05 19:19:2
*ONLINE	fab5-ep:KTM	TM	06.01.00	10/18/05 19:19:4
*ONLINE	fab5-ep:AMX	TM	06.01.00	10/18/05 19:19:4
*ONLINE	fab2-ep:KTM	TM	06.01.00	10/18/05 19:18:4
*ONLINE	fab2-ep:AMX	TM	06.01.00	10/18/05 19:18:4
*ONLINE	fab1-ep:AMX	TM	06.01.00	10/18/05 19:18:2
*ONLINE	fab1-ep:KTM	TM	06.01.00	10/18/05 19:18:2
*ONLINE	fab-ep:AMX	TM	06.01.00	10/18/05 19:18:0
*ONLINE	fab-ep:KTM	TM	06.01.00	10/18/05 19:18:0
*ONLINE	berlin-ep:KTM	TM	06.01.00	10/18/05 19:25:4
*ONLINE	berlin-ep:AMW	TM	06.01.00	10/18/05 19:25:4
*ONLINE	chicco-ep:KTM	TM	06.01.00	10/18/05 19:16:1
*ONLINE	chicco-ep:AMW	TM	06.01.00	10/18/05 19:16:1
*ONLINE	barcelona-ep:KTM	TM	06.01.00	10/18/05 19:15:5
*ONLINE	barcelona-ep:AMX	TM	06.01.00	10/18/05 19:15:5

Ok to continue?

Clearing madrid-ep:TM will also clear these offline managed system entries

```
madrid-ep:AMX
madrid-ep:KTM
```

Yes No

Physical

Hub Time: Wed, 10/19/2005 10:44 AM | Server Available | Managed System Status - kllaa9b - SYSADMIN

Figure 6-25 Clear TM offline entry confirmation dialog

6.12 Custom Resource Model support

The Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint provides data from IBM Tivoli Monitoring 5.1.2 resource models to IBM Tivoli Monitoring 6.1.

This involves mapping the data from each resource model to the appropriate tables in IBM Tivoli Monitoring 6.1. The monitoring agent includes all of the data logged by all of the IBM Tivoli Monitoring component software to be mapped to the appropriate tables. When the IBM Tivoli Monitoring 5.1.2 engine sends data to the monitoring agent, the agent finds the definition of the table to be used to send the data to the IBM Tivoli Monitoring 6.1 system.

Any data that does not match one of the known mappings is placed in the KTM_Logged_Data table. Refer to “Logged Data workspaces” on page 353 for a complete list of the information contained in this table.

All of the data logged by the IBM Tivoli Monitoring 5.1.2 engine is available in views in the Tivoli Enterprise Portal and in the warehouse. There is no loss of data if the agent does not have a custom table definition for a resource model. The advantage of the custom tables is that they provide a better organization of the data.

Refer to 6.6, “IBM Tivoli Monitoring component software supported by Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint” on page 326 for a full list of the component software that is used to generate the metadata.

Data is identified by the combination of the category, resource model name, context, and resource logged with the data. When these elements of the data match, the corresponding data is logged in the appropriate custom table. For the purpose of identifying where logged data for a resource model is to be logged, resource models are divided into three types as described in 6.12.1, “Types of resource models for logging data” on page 369.

If you are running a combination of the different types of resource models, you must know where the data is displayed in the Tivoli Enterprise Portal and where it is logged for each type of resource model.

6.12.1 Types of resource models for logging data

A resource model belongs to one of three types when considering where data for the resource model is logged:

- Resource models provided by an IBM Tivoli Monitoring component software product

  The agent includes mapping data for all of the resource models provided by IBM Tivoli Monitoring 5.1.2 and associated component software. This data represents the logging behavior of the component software products as defined in the latest version of the component software.

  Refer to 6.6, "IBM Tivoli Monitoring component software supported by Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint" on page 326 for a full list of the component software. All of the data for these resource models is logged into product-specific tables.

- Resource models based on component software resource models provided by IBM where the category, resource model name, context, and resource are the same as the shipped resource model

  Modifications to other parts of the resource model, including additional logic, calculations, and added indications do not affect the match with the mapping data that is provided. If a resource model is modified so that properties are removed from a DefineLogInst statement, the result is empty columns in the Tivoli Enterprise Portal and Warehouse. If properties are added, they are logged to the default table (but all of the original properties are logged to the custom table).

- New resource models

  Resource models that are newly developed or that are based on an IBM resource model where the category, resource model name, context, or resource has been modified. The data logged by the resource model logging statements is logged into the KTM_Logged_Data table. Refer to "Logged Data workspaces" on page 353 for a complete list of the information contained in this table.

6.12.2 Adding custom resource model support

Additional metadata can be added to the system to allow custom or modified resource models to log to custom data tables in the same way that data is logged for component software resource models provided by IBM.

The process of adding custom resource model support consists of the following major operations, details of which are described in the next section:

1. Collecting files created by the agent from the endpoint and combining them into valid definition files (which only has to be done once for all of the endpoints and custom resource models that have been defined)
2. Processing the definition files into one or more groups to produce all of the files needed by the monitoring agent and IBM Tivoli Monitoring 6.1
3. Adding the definitions to IBM Tivoli Monitoring 6.1
4. Adding the definitions to the monitoring agents
5. Restarting the IBM Tivoli Monitoring 5.1.2 engine on the affected endpoints

The definitions for the logging statements for the monitoring agent for which you are creating custom resource model support are located in the XML file for that monitoring component. For example, UNIX and Linux field definitions are in the UNIX_Linux.xml file.

Refer to the “Adding custom resource model support” section of *IBM Tivoli Monitoring: IBM Tivoli Monitoring 5.x Endpoint Agent User's Guide*, SC32-9490, for a detailed list of the applicable definitions.

6.12.3 Procedure

This section describes the detailed steps to add a custom resource model:

1. Run the unsupported resource models with all data logging enabled on an endpoint with the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint installed, configured properly and running.

   The configuration must include DataSeeding=ITM6 or DataSeeding=BOTH.
2. When the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint receives the DefineLogInst data for the unsupported resource models, the monitoring agent creates .sample files for each DefineLogInst that did not have existing metadata. These are created in the $LCF_DATDIR/LCFNEW/KTM/metadata directory.
3. Combine the .sample files as follows:
   a. Collect the .sample files and combine them into a single file called `category.xml` for each logical group of resource models.
   b. Remove any extraneous </category><category> tags when combining the .sample files. The final file must contain only one <category></category> pair.

4. Edit the following attribute elements:
   - Table names (maximum of 10 characters and uppercase only)
   - Column names (same as table name)
   - Category display name
   - String lengths, as needed
   - nodeType (three uppercase letters or numbers to denote type for category)
   - Captions for attributes

     To add a caption, specify `caption=.text.` for the <attr> tags.

   Note: The agent generates table names for you in the .sample files. If you use the generated names, be sure they are unique throughout your entire combined XML file.

5. Copy the TOOLS directory that is on the CD.
6. Use the resourceMap.properties file to map the resource value specified in the resource model to a different name in the Tivoli Enterprise Portal.

   Note: This file must be in the current directory when the ktm-seedgen tool is run.

7. Run the following command:

   ```
   java -jar ktm-seedgen.jar category.xml
   ```

   This produces the following files:

   ktm_*nodeType*.cat

   ktm_*nodeType*.atr

   ktm_*nodeType*.his

   doc*nodeType*

   ktm_*nodeType*_pres.sql

   ktm_*nodeType*_kcj.sql

   nodeType.summary
8. Run the ktm-seed-install script for the operating system that you are using.

Important:

- The scripts require that all generated files from step 7 be located in the current directory for the script.
- Check the following:
  - Windows: Verify that CANDLE_HOME is set correctly as follows:

    ```
    ktm-seed-install.bat nodeType
    ```

  - Linux or UNIX: Verify that CANDLEHOME variable is set correctly as follows:

    ```
    ktm-seed-install.sh nodeType
    ```

The scripts install the support files for the custom resource model on the Tivoli Enterprise Monitoring Server and Tivoli Enterprise Portal Server on the computer where it is being executed.

Note: If your installation contains multiple Tivoli Enterprise Monitoring Servers, or your Tivoli Enterprise Portal Server and Tivoli Enterprise Monitoring Server are located on different computers, repeat this step on each computer.

Repeat these steps on the Hot Standby TEMS, if configured, to avoid being out of sync of the TEMS catalog.

9. Restart the Tivoli Enterprise Monitoring Server and Tivoli Enterprise Portal Server to make the support files available.
10. Copy the following files to the Tivoli environment or managed node with the ITM61AGT package installed:

    category.xml

    ktm_nodeType.atr
11. Run the following command:

    ```
    witm61metadata -a path_to_category.xml path_to_ktm_nodeType.atr
    ```

    With this command you can add (-a) or remove (-r) metadata and attrlib files.

    -a The metadata file and the attrlib file must be supplied in pairs. When adding, the files must be found on the machine where the command is being run, and must be supplied with absolute paths. The command will then distribute them to every gateway in the local Tivoli management region that has the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint product installed, placing them in that gateway's LCFNEW (lcf_bundle.40) directory as shown in

Example 6-4. To make sure that the gateway cache is updated with the new dependency list, run a **wgatway *gw_name* dbcheck** against the gateways that handle Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint.

Note: If your Tivoli Management Environment contains interconnected Tivoli management regions (TMRs), you must execute the **wit61metadata** command for every TMR where the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint product is installed.

Example 6-4 Added metadata and attrlib files listing

```
metadata directory

[root@istanbul][/Tivoli/usr/local/Tivoli/bin/lcf_bundle.40/generic/TME/KTM/meta
data]-> ls -la
total 1448
drwxr-xr-x   2 root     system        1024 Oct 26 13:23 .
drwxr-xr-x   4 root     system         512 Oct 21 15:44 ..
-rwxr-xr-x   1 root     system        5651 Oct 13 16:04
Active_Directory_DC.xml
-rwxr-xr-x   1 root     system        8345 Oct 13 16:04
Active_Directory_Replication.xml
-rwxr-xr-x   1 root     system        3670 Oct 13 16:04 Apache.xml
-rwxr-xr-x   1 root     system       29611 Oct 13 16:04 CitrixMetaFrameAS.xml
-rwxr-xr-x   1 root     system       44084 Oct 13 16:04 DB2Monitoring.xml
-rwxr-xr-x   1 root     system        1551 Oct 13 16:04 DHCP.xml
-rwxr-xr-x   1 root     system        3184 Oct 13 16:04 DNS.xml
-rwxr-xr-x   1 root     system       36909 Oct 13 16:04 Domino.xml
-rwxr-xr-x   1 root     system       14892 Oct 13 16:04 IBMInformix.xml
-rwxr-xr-x   1 root     system        7376 Oct 13 16:04 IIS.xml
-rwxr-xr-x   1 root     system        5545 Oct 13 16:04 IPlanet.xml
-rwxr-xr-x   1 root     system        3776 Oct 13 16:04 MQ_Workflow.xml
-rwxr-xr-x   1 root     system       17173 Oct 13 16:04 MSBizMon.xml
-rwxr-xr-x   1 root     system        6081 Oct 13 16:04 MSCMMon.xml
-rwxr-xr-x   1 root     system       14322 Oct 13 16:04 MSCSMon.xml
-rwxr-xr-x   1 root     system       29160 Oct 13 16:04 MSHISMon.xml
-rwxr-xr-x   1 root     system       22982 Oct 13 16:04 MSISA04Mon.xml
-rwxr-xr-x   1 root     system       25151 Oct 13 16:04 MSISAMon.xml
-rwxr-xr-x   1 root     system       18171 Oct 13 16:04 MSSPPMon.xml
-rwxr-xr-x   1 root     system       11601 Oct 13 16:04 MSUDDIMon.xml
-rwxr-xr-x   1 root     system       22178 Oct 13 16:04
Microsoft_Exchange_Server.xml
-rwxr-xr-x   1 root     system       18962 Oct 13 16:04
Microsoft_SQL_Server.xml
-rwxr-xr-x   1 root     system       55465 Oct 13 16:04 Oracle.xml
-rwxr-xr-x   1 root     system        3877 Oct 13 16:04 Resource_Model.xml
-rwxr-xr-x   1 root     system       13162 Oct 13 16:04 Siebel.xml
```

```
-rwxr-xr-x   1 root     system         2711 Oct 13 16:04 Solaris.xml
-rwxr-xr-x   1 root     system        10225 Oct 13 16:04 SybaseMon.xml
-rwxr-xr-x   1 root     system        11353 Oct 13 16:04 UNIX_Linux.xml
-rwxr-xr-x   1 root     system         8223 Oct 13 16:04 VMWareMon.xml
-rwxr-xr-x   1 root     system         5861 Oct 13 16:04 WICS.xml
-rwxr-xr-x   1 root     system        15963 Oct 13 16:04 Weblogic.xml
-rwxr-xr-x   1 root     system        22631 Oct 13 16:04 WebsphereAS.xml
-rwxr-xr-x   1 root     system        53502 Oct 13 16:04 Websphere_MQ.xml
-rwxr-xr-x   1 root     system        33689 Oct 13 16:04 Websphere_MQI.xml
-rwxr-xr-x   1 root     system        20610 Oct 13 16:04 Windows.xml

Customized files start here

-rw-r-----   1 root     system          414 Oct 25 13:57 category.xml
-rw-r--r--   1 root     system          558 Oct 26 12:02 fab.xml
-rw-rw-rw-   1 root     system          419 Oct 25 17:51 fabrizio.xml
-rwxr-xr-x   1 root     system        35319 Oct 13 16:04 mySAP.xml
[root@istanbul][/Tivoli/usr/local/Tivoli/bin/lcf_bundle.40/generic/TME/KTM/meta
data]->

ATTRLIB directory

[root@istanbul][/Tivoli/usr/local/Tivoli/bin/lcf_bundle.40/generic/TME/KTM/ATTR
LIB]-> ls -la
total 2768
drwxr-xr-x   2 root     system         1024 Oct 26 13:23 .
drwxr-xr-x   4 root     system          512 Oct 21 15:44 ..
-rwxr-xr-x   1 root     system        29907 Oct 13 16:04 kib.atr
-rwxr-xr-x   1 root     system         7994 Oct 13 16:04 ktm.atr
-rwxr-xr-x   1 root     system        75632 Oct 13 16:04 ktm_aba.atr
-rwxr-xr-x   1 root     system        72912 Oct 13 16:04 ktm_abh.atr
-rwxr-xr-x   1 root     system        12739 Oct 13 16:04 ktm_ama.atr
-rwxr-xr-x   1 root     system        18930 Oct 13 16:04 ktm_amb.atr
-rwxr-xr-x   1 root     system         3086 Oct 13 16:04 ktm_amd.atr
-rwxr-xr-x   1 root     system         7049 Oct 13 16:04 ktm_amn.atr
-rwxr-xr-x   1 root     system         6273 Oct 13 16:04 ktm_ams.atr
-rwxr-xr-x   1 root     system        46210 Oct 13 16:04 ktm_amw.atr
-rwxr-xr-x   1 root     system        25020 Oct 13 16:04 ktm_amx.atr
-rwxr-xr-x   1 root     system         7229 Oct 13 16:04 ktm_biw.atr
-rwxr-xr-x   1 root     system        11500 Oct 13 16:04 ktm_bix.atr
-rwxr-xr-x   1 root     system        86395 Oct 13 16:04 ktm_ctd.atr
-rwxr-xr-x   1 root     system       109512 Oct 13 16:04 ktm_cto.atr
-rwxr-xr-x   1 root     system       107056 Oct 13 16:04 ktm_ctq.atr
-rwxr-xr-x   1 root     system        30235 Oct 13 16:04 ktm_ctr.atr
-rwxr-xr-x   1 root     system        37643 Oct 13 16:04 ktm_ctw.atr
-rwxr-xr-x   1 root     system        44027 Oct 13 16:04 ktm_cty.atr
-rwxr-xr-x   1 root     system        26071 Oct 13 16:04 ktm_gms.atr
-rwxr-xr-x   1 root     system         7530 Oct 13 16:04 ktm_gwa.atr
-rwxr-xr-x   1 root     system        15790 Oct 13 16:04 ktm_gwi.atr
```

```
-rwxr-xr-x   1 root     system        30281 Oct 13 16:04 ktm_gwl.atr
-rwxr-xr-x   1 root     system        11276 Oct 13 16:04 ktm_gwp.atr
-rwxr-xr-x   1 root     system        28580 Oct 13 16:04 ktm_iqs.atr
-rwxr-xr-x   1 root     system        51151 Oct 13 16:04 ktm_iqy.atr
-rwxr-xr-x   1 root     system        35550 Oct 13 16:04 ktm_iqz.atr
-rwxr-xr-x   1 root     system        24233 Oct 13 16:04 ktm_iud.atr
-rwxr-xr-x   1 root     system        58172 Oct 13 16:04 ktm_iui.atr
-rwxr-xr-x   1 root     system        47683 Oct 13 16:04 ktm_ivd.atr
-rwxr-xr-x   1 root     system        16767 Oct 13 16:04 ktm_ivi.atr
-rwxr-xr-x   1 root     system        35765 Oct 13 16:04 ktm_ixa.atr
-rwxr-xr-x   1 root     system        22431 Oct 13 16:04 ktm_ixb.atr
-rwxr-xr-x   1 root     system        59524 Oct 13 16:04 ktm_ixt.atr
-rwxr-xr-x   1 root     system        11973 Oct 13 16:04 ktm_iym.atr
-rwxr-xr-x   1 root     system        41997 Oct 13 16:04 ktm_izy.atr
-rwxr-xr-x   1 root     system        65292 Oct 13 16:04 ktm_mmi.atr

Customized files start here

-rw-r-----   1 root     system          931 Oct 25 13:57 ktm_om.atr
-rw-rw-rw-   1 root     system         1194 Oct 26 13:06 ktm_omt.atr
-rw-rw-rw-   1 root     system         1002 Oct 25 17:51 ktm_ppp.atr
```

-r Remove is similar, except that the files are specified without absolute paths; just the base filename should be provided. The remove only removes the files from the depset but leaves them on the Gateways.

Note: To be sure the command completed successfully, run the **`witm61metadata`** command with no parameters. This dumps the list of metadata and attrlib files (respectively .xml and .atr) that are associated to the ktm_metadata dependency set.

The new files are added at the bottom of the list.

12. Run the **`witm61agt`** command to push the new .xml and .atr files to the endpoints.

Note: You do not have to run the command with the -i flag to reset the DataSeeding because it will not be lost.

When the command completes, check that the new files have been successfully downloaded on the right path on the Endpoint:

```
$LCF_DATDIR/LCFNEW/KTM/metadata for the .xml file
$LCF_DATDIR/LCFNEW/KTM/ATTRLIB for the .atr file
```

13.Restart the IBM Tivoli Monitoring 5.1.2 engine on the endpoints with the new metadata in place. The custom resource model data is now displayed under its own group instead of under the default Logged Data.

6.12.4 Custom Resource Model implementation example

This section shows some commands and screen shots from a real custom resource model implementation in the environment used for this book.

Example 6-5 shows an example of some the procedure described in 6.12.3, “Procedure” on page 370.

Example 6-5 Custom resource model integration

```
Steps executed at the TMR server side:

1) copy the customer resource model tar file on a directory in the TMR server
machine, and load it in the TMR database:

[root@istanbul][/tmp/customrm/ITM61_Custom]-> wdmrm -add ITM61_Custom.tar

IBM Tivoli Monitoring - Adding new resource model

Parsing configuration file OpenESM_Services.conf ...
Configuration file successfully parsed.
Checking for event redefinition...
Starting resource OpenESM_Services registration ...
The resource OpenESM_Services has been successfully stored.
Registration completed.

Copying OpenESM_Services.cat msgfile ...
Copying OpenESM_Services.cat zipfile ...

Installation completed.

2) configure a Tmw2kProfile to add the custom resource model, then make all the
required customization and distribute the profile to the Endpoint.

3) make sure the custom resource model is running:

[root@istanbul][/tmp/customrm/ITM61_Custom]-> wdmlseng -e chicco-ep

Forwarding the request to the endpoint:
chicco-ep  1567175343.5.522+#TMF_Endpoint::Endpoint#

The following profiles are running:

WINDOWS_UD#istanbul-region
```

```
    TMW_ParamServices        :Running
WINDOWS#istanbul-region
    TMW_NetworkIntCard       :Running
    TMW_ParamServices        :Running
    TMW_PhysicalDiskModel    :Running
    TMW_TCPIP                :Running
    TMW_LogicalDisk          :Missed Prereq
    TMW_Processor            :Running
CUSTOM_WINDOWS#istanbul-region
    OpenESM_Disk             :Running
CUSTOM#istanbul-region
    OpenESM_Services         :Running
[root@istanbul][/tmp/customrm/ITM61_Custom]->
```

Steps executed at Endpoint running the custom resource model side:

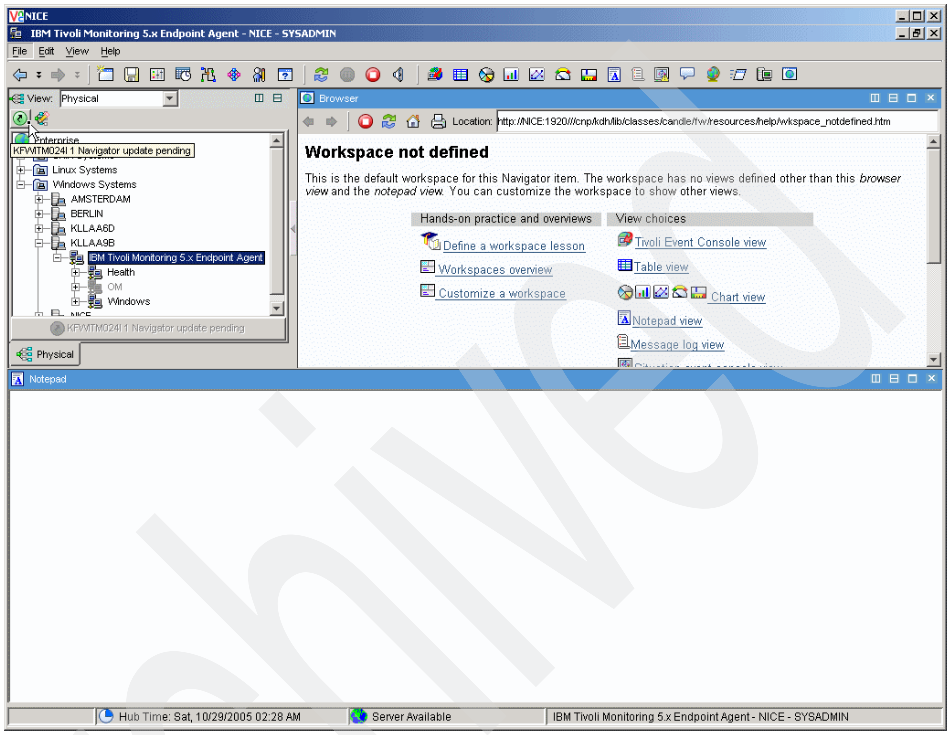
```
1) The .sample file for the custom resource model has been created and it looks
like the following:

<category name="OpenESM_Models" nodeType="CUS">
<attributeGroup object="OpenESM_Process_Windows_Process_Process"
table="OMWPPRO" ctx="Process" rm="OpenESM_Process" rsc="Windows_Process">
  <attr name="name" column="PRC_NAME" key="true" atomic="true" primarykey="0"
type="S" length="256" />
  <attr name="processcount" column="PRC_COUNT" type="I" length="4"
behavior="count" />
  </attributeGroup>
  </category>

2) modify some entry if you want different info shown

3) use the ktm-seedgen.jar command to create all the custom resource model
files for the IBM Tivoli Monitoring 6.1 integration

C:\Program Files\Tivoli\lcf\dat\1\LCFNEW\KTM\metadata\custom>java -jar
ktm-seedgen.jar ITM61_Custom.xml
Category: OpenESM_Models, nodeType CUS, display OpenESM_Models
Attribute group OpenESM_Services_Windows_Service_Status
Creating file cus_kcj.sql
Creating file cus_pres.sql
Resource Windows_Service has 1 tables
Generating CAT/ATR/HIS files

C:\Program Files\Tivoli\lcf\dat\1\LCFNEW\KTM\metadata\custom>dir
 Volume in drive C has no label.
 Volume Serial Number is 046B-C174

 Directory of C:\Program Files\Tivoli\lcf\dat\1\LCFNEW\KTM\metadata\custom
```

```
10/29/2005  02:40a      <DIR>          .
10/29/2005  02:40a      <DIR>          ..
10/29/2005  02:40a                  70 cus.summary
10/29/2005  02:40a                 931 doccus
10/29/2005  02:35a                 669 ITM61_Custom.xml
10/13/2005  04:09p             102,344 jcatnip.jar
10/13/2005  04:09p             455,474 kjrall.jar
10/13/2005  04:09p               1,313 ktm-seed-install.bat
10/13/2005  04:09p               1,632 ktm-seed-install.sh
10/13/2005  04:09p              34,589 ktm-seedgen.jar
10/29/2005  02:40a               1,571 ktm_cus.atr
10/29/2005  02:40a               1,987 ktm_cus.cat
10/29/2005  02:40a               1,000 ktm_cus.his
10/29/2005  02:40a               4,172 ktm_cus_kcj.sql
10/29/2005  02:40a               3,770 ktm_cus_pres.sql
10/13/2005  04:09p             853,811 tap_cli.jar
10/13/2005  04:09p             176,892 util.jar
              15 File(s)      1,640,225 bytes
               2 Dir(s)   9,831,952,384 bytes free

Steps executed at the TEMS/TEPS side:

1) Seed the custom resource model files to the TEMS and TEPS.

C:\custom\ITM61_Custom>ktm-seed-install.bat cus
Installing TEMS support
        1 file(s) copied.
        1 file(s) copied.
Installing TEPS support
        1 file(s) copied.
        1 file(s) copied.
        1 file(s) copied.
        1 file(s) copied.
ktm_cus_kcj.sql
ktm_cus_pres.sql
        2 file(s) copied.
Seeding TEPS
Please restart TEMS and TEPS for changes to take effect

Note:In case the TEMS and TEPS are not on the same machine, run this command
for each system. The same applies in case you have multiple HUB TEMS or TEPS.

Steps executed at TMR server side:

1) Copy all the files generated by the kmt-seedgen.jar command and all the
other jar files present in the TOOLS directory of the installation CD and put
them in a directory on the TMR server.

2) add the files to the ktm_metadata dependency set:
```

```
[root@istanbul][/tmp/customrm/ITM61_Custom]-> witm61metadata -a
/tmp/customrm/ITM61_Custom/ITM61_Custom.xml
/tmp/customrm/ITM61_Custom/ktm_cus.atr
KTM0013I Distribution of new files to gateway istanbul: SUCCEEDED.

3) run a gateway dbcheck to update the dependency set:

[root@istanbul][/tmp/customrm/ITM61_Custom]-> wgateway istanbul-gw dbcheck

4) run again the witm61agt command to push the new .xml amd .atr files on the
Endpoint

[root@istanbul][/tmp/customrm/ITM61_Custom]-> witm61agt chicco-ep -c nice -p
1958
KTM0009I Endpoint chicco-ep: SUCCEEDED

5) restart the IBM Tivoli Monitoring engine to restart the ktmcma process:

[root@istanbul][/tmp/customrm/ITM61_Custom]-> wdmcmd -e chicco-ep -restart
Starting engine on endpoint 1567175343.5.522+#TMF_Endpoint::Endpoint#
Started engine on endpoint 1567175343.5.522+#TMF_Endpoint::Endpoint#
[root@istanbul][/tmp/customrm/ITM61_Custom]->
```

After all steps have been executed, you can open the TEP to see your custom resource model data as shown in the following figures.

- TEP navigator update pending icon


NICE
IBM Tivoli Monitoring 5.x Endpoint Agent - NICE - SYSADMIN
File
Edit
View
Help
View: Physical
Enterprise
KFWITM024I 1 Navigator update pending
Linux Systems
Windows Systems
AMSTERDAM
BERLIN
KLLAA6D
KLLAA9B
IBM Tivoli Monitoring 5.x Endpoint Agent
Health
OM
Windows
NICE
KFWITM024I 1 Navigator update pending
Physical
Browser
Location: http://NICE:1920///cnp/kdh/lib/classes/candle/fw/resources/help/wkspace_notdefined.htm
Hands-on practice and overviews
Define a workspace lesson
Workspaces overview
Customize a workspace
View choices
Tivoli Event Console view
Table view
Chart view
Notepad view
Message log view
Situation event console view
Notepad
Hub Time: Sat, 10/29/2005 02:28 AM
Server Available
IBM Tivoli Monitoring 5.x Endpoint Agent - NICE - SYSADMIN

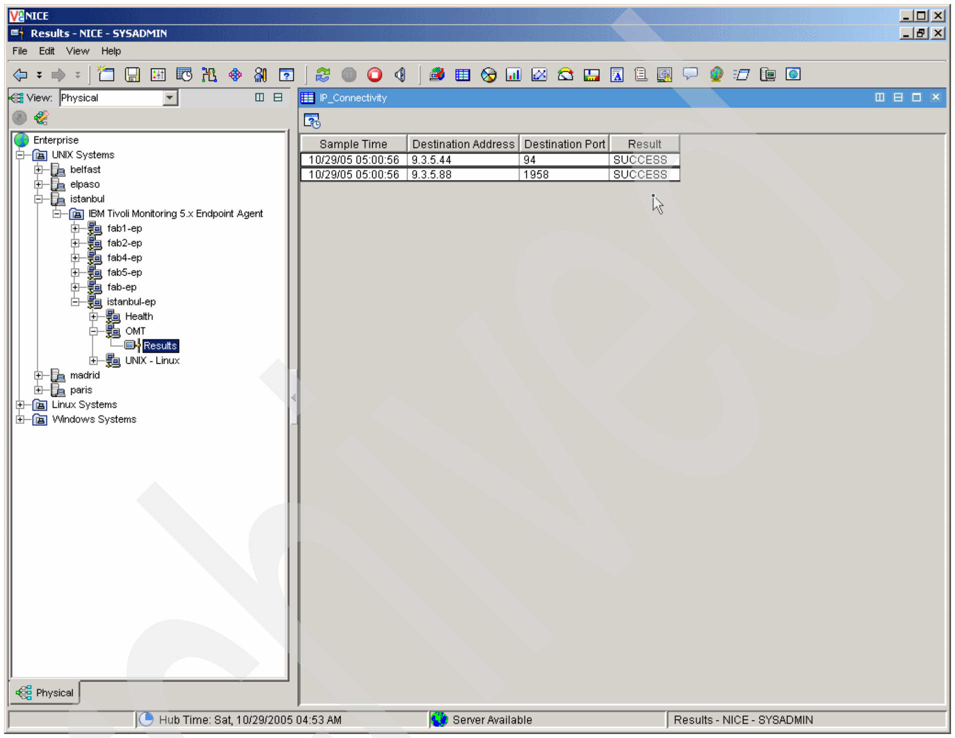
Workspace not defined

This is the default workspace for this Navigator item. The workspace has no views defined other than this *browser view* and the *notepad view*. You can customize the workspace to show other views.

Figure 6-26 TEP navigator update pending icon

- Custom resource model view

  The custom resource model category defined in the XML file is shown as a subtree of the IBM Tivoli Monitoring 5.x Endpoint Agent subtree for the specific Agent as shown in Figure 6-27.



Sample Time	Destination Address	Destination Port	Result
10/29/05 05:00:56	9.3.5.44	94	SUCCESS
10/29/05 05:00:56	9.3.5.88	1958	SUCCESS

Figure 6-27 Custom resource model view

6.12.5 Under the covers

This section provides some “under the covers” information:

- Seed generator will make some simple validations on the supplied XML file to avoid problems and enforce limitations.
- Seed generator will create ODI file, CAT, ATR, and HIS files based on the XML metadata file (XSL transform followed by running JCatnip on the resulting ODI file).
- A query will be created for each table.
- Tables are grouped by resource used in the defineLogInst calls.

- Workspaces will be generated containing one to four views, depending on how many queries are associated with a given resource.
- Customization of resource groupings, resource names, resource hiding, and workspaces controlled via property files (refer to 6.12.6, “Customizing the generator” on page 382)

6.12.6 Customizing the generator

These are the property files that control the generator:

- rscRscMap.properties

  Maps resource names into different resource names. Can be used to group resources together or to rename resources in order to change display order in the navigation tree or to normalize resource names used in different resource models. For example:

  ```
  DB2Apply=DB2Replication
  ```

- resourceMap.properties

  Maps the internal resource name used in the resource model to a human-friendly name. For example:

  ```
  DB2Replication=Replication
  ```

- hideIds.properties

  Hides some resources from the navigation tree. Used to cut unnecessary or irrelevant clutter from tree. For example:

  ```
  zktm_AMX_RealSystem=true
  ```

6.12.7 Metadata files

The metadata files for all IBM-shipped resource models (base IBM Tivoli Monitoring and PACs) are provided. All of these metadata files are sent to the Endpoint where you install the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint during the execution of the **witm61agt** command.

The content of the metadata is based primarily on the data stored in the resource model. Each DefineLogInstEx statement results in a unique table. The table name is:

CategoryName-RMName-ContextName-ResourceName

The metadata files contain the IRA table and column names with the appropriate length limits. In addition, the table name is prefixed with the three-letter nodeType to make it unique when data warehousing is enabled.

The attributes in each attribute group are defined by the list of attributes being logged and the order they are listed in the DefineLogInstEx statement following these guidelines:

- Key attributes
- Numeric attributes
- String attributes

Strings have a 256-byte default.

In the case of *numeric* data, the agent tries to guess the metric name by looking at some patterns, such as pct, Percent, Avg, Average, and so on, and decide whether scaling should be used. Scaling should be used carefully as it reduces the value range that can be handled. Numeric data uses the new capabilities in IBM Tivoli Monitoring 6.1. The scale attribute of a column determines the multiplier applied by the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint that transforms the number received from the IBM Tivoli Monitoring engine (which is always a floating point number) into a fixed-point decimal.

Keyed data, based on the keys in the logging, are used, so the IBM Tivoli Monitoring engine feeds to the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint update the data in memory instead of adding another copy.

Example 6-6 shows an example of a metadata file.

Example 6-6 Extract of a UNIX_Linux metadata file

```
<category name="UNIX_Linux" nodeType="AMX" displayName="UNIX - Linux"
interpList="-w32-ix86 -os400">
    <attributeGroup object="DMXCpu_Availability_RealSystem" table="AMXCPUAVBR"
ctx="Availability" rm="DMXCpu" rsc="RealSystem">
        <attr name="Name" column="NAME" type="S" length="256" atomic="true"
key="true" primarykey="0"/>
        <attr name="SystemUpTime" column="SYSTEMUPTI" type="I" length="4"
behavior="gauge" option="TYPE=kfw.TimeCounter"/>
    </attributeGroup>
<attributeGroup object="DMXCpu_Average_Loading_CPU" table="AMXPAVGLDG"
ctx="Average Loading" rm="DMXCpu" rsc="CPU">
        <attr name="name" column="NAME" type="S" length="256" atomic="true"
key="true" primarykey="0"/>
        <attr name="loadAvg1" column="LOADAVG1" type="I" length="4"
precision="12" scale="2" behavior="gauge"/>
        <attr name="loadAvg5" column="LOADAVG5" type="I" length="4"
precision="12" scale="2" behavior="gauge"/>
        <attr name="loadAvg15" column="LOADAVG15" type="I" length="4"
precision="12" scale="2" behavior="gauge"/>
    </attributeGroup>
</category>
```

6.13 Problem determination

“Appendix F. Problem Determination” of *IBM Tivoli Monitoring: IBM Tivoli Monitoring 5.x Endpoint Agent User's Guide*, SC32-9490, contains a detailed description of how to troubleshoot the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint.

Because the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint provides integration between IBM Tivoli Monitoring 5.x and IBM Tivoli Monitoring 6.x, parts of the monitoring agent run within Version 5.1.2 and parts run within Version 6.x.

You can obtain a complete view of the state of the agent by using the problem determination features of each system:

- For problems related to the display of the data in the Tivoli Enterprise Portal, and storing the data in the warehouse, see the problem determination features of IBM Tivoli Monitoring 6.x.
- For problems related to gathering the data, see the problem determination features of IBM Tivoli Monitoring 5.x, in conjunction with the Version 6.x agent problem determination features.

This appendix provides agent-specific problem determination information. See the IBM Tivoli Monitoring Problem Determination Guide for general problem determination information.

Also see “Support information” at the end of Appendix F for detailed information about the use of the knowledge base, obtaining fixes, receiving weekly support updates, and contacting IBM Software Support.

6.13.1 Common information to look at

For an initial determination of any problem, check the following:

- Look at the agent logs under these files:

  $LCF_DATDIR/LCFNEW/KTM/logs/<*ep_name*>_<*timestamp*>-<*n*>.log
- Look at the messages workspace for any problems surfaced by the agent.
- Increase log level if no source for the problem found.
- Turn on debugging of the IBM Tivoli Monitoring 5.x stream. (Add UNIT:server INPUT to the KTMENV file and restart the IBM Tivoli Monitoring engine.)
- Log IBM Tivoli Monitoring 5.x stream to ktmout.txt by adding KTM_DEBUG_INPUT=Y to KTMENV, and restart the IBM Tivoli Monitoring engine.

- Look at the data cache traces. (Add UNIT:datacache ALL to the KTMENV file and restart the IBM Tivoli Monitoring engine.) Will show most hidden errors that could grow log size and scavenger operation.
- Look at the data being emitted to the IRA. (Add UNIT:genericagent OUTPUT to the KTMENV file and restart the IBM Tivoli Monitoring engine.)
- Check the IBM Tivoli Monitoring engine trace log files for any problem in the pipe between the two processes.

6.13.2 Problem areas

This section lists some of the most common problem areas, and for each one of them, provides some information about the possible main cause of failure:

- Problem: data appears and disappears in TEP
  - Likely cause is that some resource model is taking too much time when running, causing others to be delayed beyond 2.5 times their cycle times.
  - Use either `KTM_DEBUG_INPUT=Y` in the KTMENV to see the exact stream coming from IBM Tivoli Monitoring 5.x, or enable the Server.cpp trace (UNIT:server INPUT).
  - Enable the detail-level tracing in the DataCache.cpp (UNIT:datacache DETAIL METRICS). Shows scavenger operation in removing expired rows.
- Problem: data not being exported
  - Make sure the ATR file in the $LCF_DATDIR/LCFNEW/KTM/ATTRLIB directory for the data not being exported is not in DOS format on UNIX machines. (Using vi, there should not be ^M at the end of lines.)
  - If binary files are being created in the hist directory (<table_name> and <table_name>.hdr), make sure Warehouse Proxy has IP.PIPE or IP.UDP protocol available. The endpoint agent does not support SSL connections.
- Problem: agent not running
  - Check the data seeding on the endpoint: **`wdmepconfig –e <endpoint> -G DataSeeding`**. It should be set to either BOTH or ITM6 for the agent to be started.

Example 6-7 Checking seeding information

```
[root@istanbul][/opt/Tivoli/lcf/dat/1/LCFNEW/KTM/logs]-> wdmepconfig -e
istanbul-ep -G DataSeeding
Endpoint Label: istanbul-ep

=======================
        DataSeeding="BOTH"
```

  - DataSeeding=ITM5 is the default. Setting is persisted in the LCFNEW/Tmw2k directory, so deleting the directory resets the value to the default.
  - Set up the LCF environment (lcf_env.[sh|cmd]) and try to run the agent manually. If there are no messages about missing or mismatched libraries, the agent should work correctly. Try pasting the text in Example 6-8 into the STDIN and press Enter. (This should cause the agent to show in TEP if it was not there before.)

Example 6-8 Paste this text into the STDIN

```
<logInstEx version="1.0"><sampletimestamp time="1118431756"/><category
name="UNIX_Linux"/><rm name="DMXCpu"/><context name="Availability"/><rsc
name="RealSystem"/><nAttr name="SystemUpTime">1.2350321E7</nAttr><sAttr
name="Name">System</sAttr></logInstEx>
```

    Type `^Z` on Windows or `^D` on UNIX to shut down the agent cleanly.

  - Check the IBM Tivoli Monitoring engine trace and logs for abnormal problems.

- Problem: too much warehouse data generated
  - Change the warehousing done by the agent from all received rows to the standard IBM Tivoli Monitoring 6.1 interval warehousing by adding `KTM_WAREHOUSE=INTERVAL` to KTMENV and restarting the IBM Tivoli Monitoring engine
  - This will make the agent behave like any IBM Tivoli Monitoring 6.1 agent in its warehousing and will allow the usage of the WRITETIME column as the default category
  - Warehouse mode being used is printed to the RAS1 log
- Problem: agent running, but cannot talk to TEMS
  - Ensure any firewall between agent and TEMS has port 1918 (or chosen port number if different) opened for communication between agent and TEMS
  - If agent and TEMS are in different networks, a partition file may be needed, just like any agent. (See the user guide's Problem Determination appendix.)
- Problem: subnodes show up, but no report nodes
  - Verify that the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint support has been added on the TEPS. Without it, subnodes will not have any report nodes.
  - If this is a custom resource model, ensure that the SQL files have been installed in the TEPS server and have been correctly installed in the TEPS

database. (Check the KFWQUERY table for the missing entries such as `zktm_<node_type>_<resource>`). Example query ID: `zktm_AMX_CPU`

6.13.3 Service Console

To access the built-in Service Console, type `http://<endpoint_name>:1920` into a browser. Click ***<endpoint_name>*_tm** to access and provide the user and password on the endpoint machine.

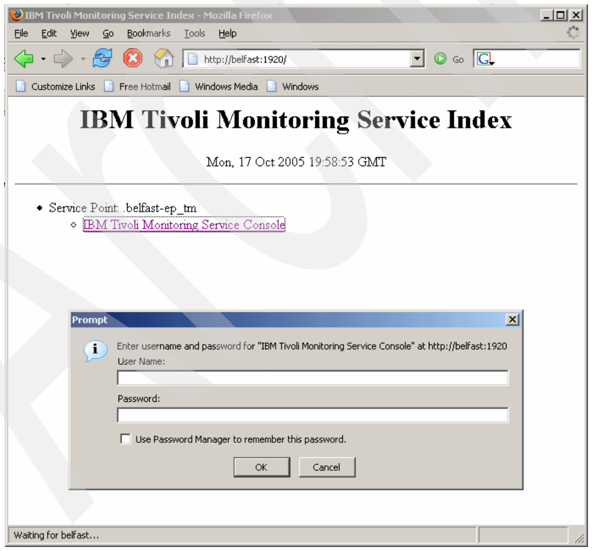
The Service Console allows access to agent log, environment variables, trace configuration, active tables and requests, and so forth.

The **`ktm`** command provides a glimpse of the agent, even if not connected to TEMS:

- Metrics about the parser, metadata, memory usage
- Current cycle times being used to expire old data
- Messages currently in the messages table

The following example shows some available Service Console options for the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint.

When you connect to port 1920, you are first requested to log on using an OS account on the machine you are connecting to, as shown in Figure 6-28.


IBM Tivoli Monitoring Service Index - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://belfast:1920/
Go
Customize Links
Free Hotmail
Windows Media
Windows

IBM Tivoli Monitoring Service Index

Mon, 17 Oct 2005 19:58:53 GMT

- Service Point: .belfast-ep_tm
  - IBM Tivoli Monitoring Service Console

Prompt

Enter username and password for "IBM Tivoli Monitoring Service Console" at http://belfast:1920

User Name:

Password:

Use Password Manager to remember this password.

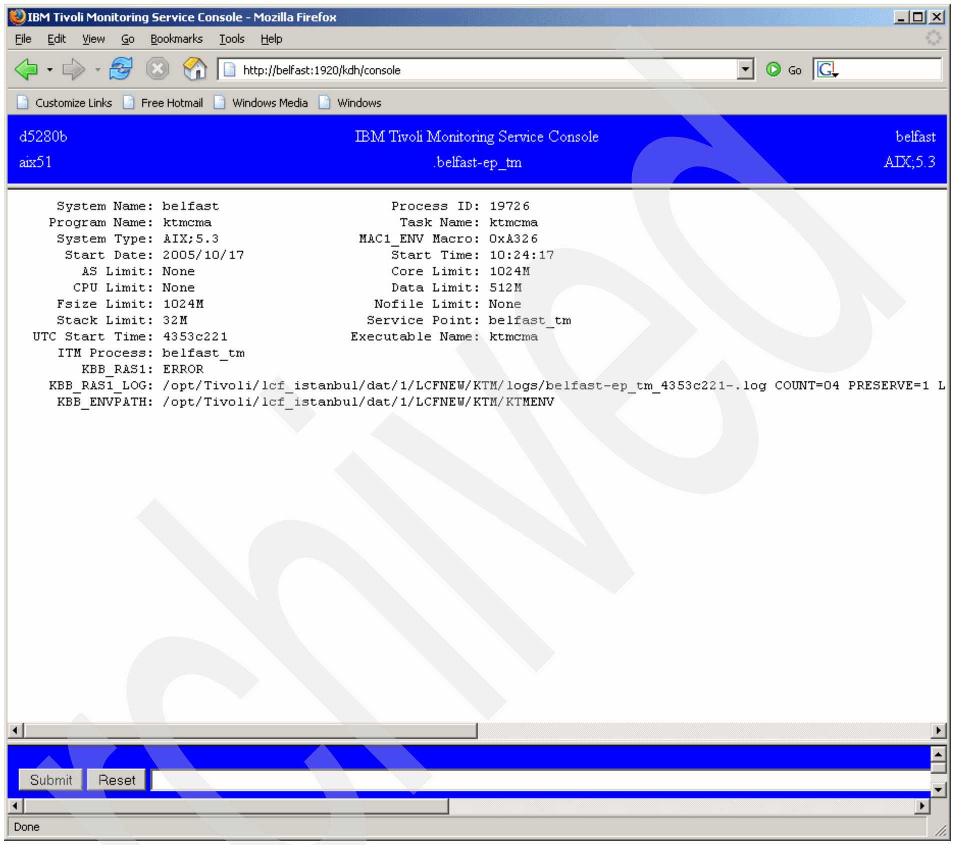
OK Cancel

Waiting for belfast...

Figure 6-28 Service Console login dialog

You should see a page similar to Figure 6-29.

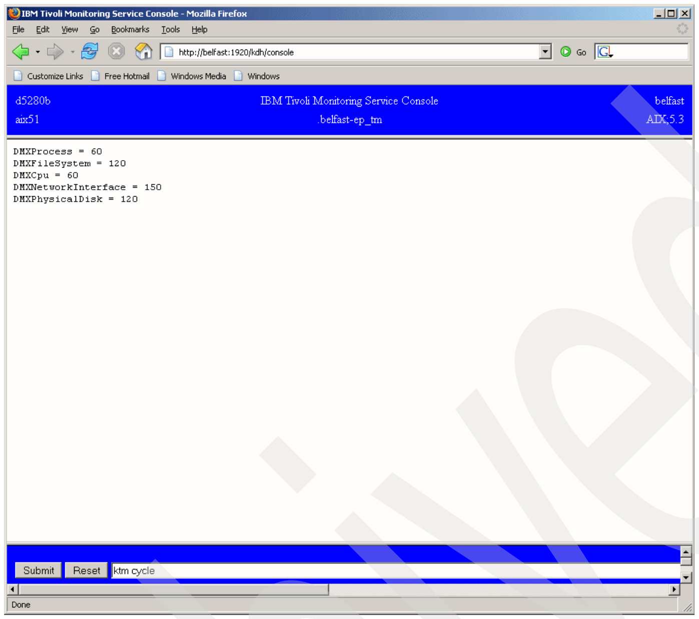
This page provides some basic information about the agent configuration, the log file name, and a command bar you can you use to run the **`ktm`** command.


IBM Tivoli Monitoring Service Console - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://belfast:1920/kdh/console
Go
Customize Links
Free Hotmail
Windows Media
Windows
Submit
Reset
Done

```
d5280b                 IBM Tivoli Monitoring Service Console                 belfast
aix51                              .belfast-ep_tm                            AIX;5.3

     System Name: belfast                        Process ID: 19726
    Program Name: ktmcma                          Task Name: ktmcma
     System Type: AIX;5.3                    MAC1_ENV Macro: 0xA326
      Start Date: 2005/10/17                     Start Time: 10:24:17
        AS Limit: None                           Core Limit: 1024M
       CPU Limit: None                           Data Limit: 512M
     Fsize Limit: 1024M                        Nofile Limit: None
     Stack Limit: 32M                         Service Point: belfast_tm
  UTC Start Time: 4353c221                  Executable Name: ktmcma
     ITM Process: belfast_tm
        KBB_RAS1: ERROR
    KBB_RAS1_LOG: /opt/Tivoli/lcf_istanbul/dat/1/LCFNEW/KTM/logs/belfast-ep_tm_4353c221-.log COUNT=04 PRESERVE=1 L
     KBB_ENVPATH: /opt/Tivoli/lcf_istanbul/dat/1/LCFNEW/KTM/KTMENV
```

Figure 6-29 Remote Console welcome page

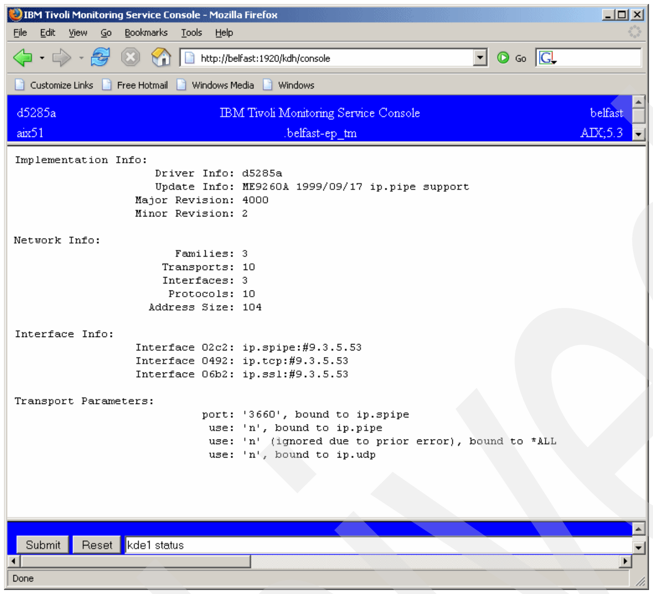
Figure 6-30 shows the output of the **`ktm cycle`** command run from the command bar of your browser session.


IBM Tivoli Monitoring Service Console - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://belfast:1920/kdh/console
Go
Customize Links Free Hotmail Windows Media Windows
d5280b
aix51
IBM Tivoli Monitoring Service Console
.belfast-ep_tm
belfast
AIX;5.3
Submit Reset
ktm cycle
Done

```
DMXProcess = 60
DMXFileSystem = 120
DMXCpu = 60
DMXNetworkInterface = 150
DMXPhysicalDisk = 120
```

Figure 6-30 ktm cycle

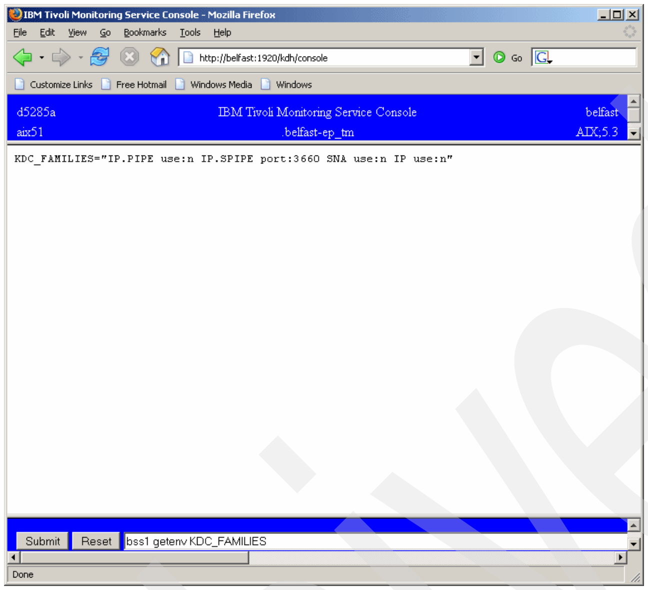
Figure 6-31 shows the output of the **kde1 status** command run from the command bar of your browser session.


IBM Tivoli Monitoring Service Console - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://belfast:1920/kdh/console
Go
Customize Links
Free Hotmail
Windows Media
Windows
d5285a
IBM Tivoli Monitoring Service Console
belfast
aix51
.belfast-ep_tm
AIX;5.3
Submit
Reset
kde1 status
Done

```
Implementation Info:
                     Driver Info: d5285a
                     Update Info: ME9260A 1999/09/17 ip.pipe support
                  Major Revision: 4000
                  Minor Revision: 2

Network Info:
                        Families: 3
                      Transports: 10
                      Interfaces: 3
                       Protocols: 10
                    Address Size: 104

Interface Info:
                  Interface 02c2: ip.spipe:#9.3.5.53
                  Interface 0492: ip.tcp:#9.3.5.53
                  Interface 06b2: ip.ssl:#9.3.5.53

Transport Parameters:
                            port: '3660', bound to ip.spipe
                             use: 'n', bound to ip.pipe
                             use: 'n' (ignored due to prior error), bound to *ALL
                             use: 'n', bound to ip.udp
```

Figure 6-31 kde1 status

Figure 6-32 shows the output of the **`bss1 getenv`** ***`variable_name`*** command run from the command bar of your browser session.

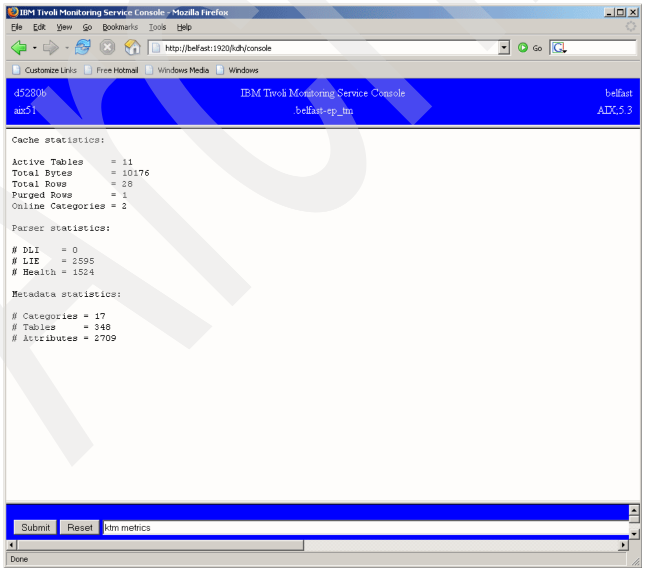

IBM Tivoli Monitoring Service Console - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://belfast:1920/kdh/console
Go
Customize Links
Free Hotmail
Windows Media
Windows
d5285a
IBM Tivoli Monitoring Service Console
belfast
aix51
.belfast-ep_tm
AIX;5.3
Submit
Reset
Done

```
KDC_FAMILIES="IP.PIPE use:n IP.SPIPE port:3660 SNA use:n IP use:n"
```

```
bss1 getenv KDC_FAMILIES
```

Figure 6-32 bss1 getenv variable_name

Figure 6-33 shows the **`ktm metric`** command output.


IBM Tivoli Monitoring Service Console - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://belfast:1920/kdh/console
Go
Customize Links
Free Hotmail
Windows Media
Windows
d5280b
IBM Tivoli Monitoring Service Console
belfast
aix51
.belfast-ep_tm
AIX;5.3
Submit
Reset
Done

```
Cache statistics:

Active Tables     = 11
Total Bytes       = 10176
Total Rows        = 28
Purged Rows       = 1
Online Categories = 2

Parser statistics:

# DLI    = 0
# LIE    = 2595
# Health = 1524

Metadata statistics:

# Categories = 17
# Tables     = 348
# Attributes = 2709
```

```
ktm metrics
```

Figure 6-33 ktm metrics

6.14 Hints and tips

This section offers some hints and tips to keep in mind while dealing with Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint.

- Deleting LCFNEW and redistributing the profiles requires:
  - Resetting the DataSeeding for the endpoint, as the configuration value will be lost and will reset to the default value of ITM5.

    Use the **`wdmepconfig`** command line to reset the DataSeeding.
  - Redistributing and configuring the endpoint agent via the **`witm61agt`** command.
- Deleting Tmw2k directory under $LCF_DATDIR/LCFNEW requires:
  - Resetting the DataSeeding for the endpoint, as mentioned above.

6.14.1 Manually running the agent

To troubleshoot the agent, you can run the ktmcma manually.

- Files needed:
  - ktmcma executable
  - metadata directory contents under metadata directory
  - ATTRLIB directory contents
  - Shared libraries or DLLs as appropriate and in the operating system appropriate path
  - Logs directory created
  - KTMENV file with CT_CMSLIST, KDC_FAMILIES, KTM_META_PATH, ATTRLIB, KBB_RAS1, KBB_RAS1_LOG, CTIRA_HIST_DIR, CTIRA_LOG_PATH, CTIRA_SIT_PATH (sample file in the next slide)
  - Set INTERP variable as needed to load correct metadata set
- If $LCF_DATDIR is not set, agent uses current directory for ktmout.txt and ktmerr.txt; otherwise, files go the under $LCF_DATDIR/KTM/logs directory.
- Manually run the agent: **`ktmcma.exe`** (or appropriate for OS).

The Example 6-9 shows a sample KTMENV file that can be used to manually run the ktmcma on a Windows OS.

Example 6-9 Sample KTMENV file for Windows OS

```
CT_CMSLIST=IP.PIPE:temshostname
KDC_FAMILIES=IP.PIPE PORT:1918 IP use:n SNA use:n IP.SPIPE use:n
CTIRA_HIST_DIR=c:\temp\candle\logs
```

```
CTIRA_LOG_PATH=c:\temp\candle\logs
CTIRA_SIT_PATH=c:\temp\candle
KTM_META_PATH=c:\temp\candle\metadata
ATTRLIB=c:\temp\candle\ATTRLIB
KBB_VARPREFIX=$
KBB_RAS1=ERROR
KBB_RAS1_LOG=$CTIRA_LOG_PATH\ktmras1.log
```

6.15 Monitoring Agent behind the scenes

This paragraph provides some information about the internals of the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint.

The Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint is implemented using a generic IRA agent based on the metadata files discussed in 6.12.7, “Metadata files” on page 382. It reads from the STDIN file descriptor for the data flow from the IBM Tivoli Monitoring engine. The IBM Tivoli Monitoring engine is responsible for starting and stopping the agent, so the agent runs under the same account used to run the engine (typically root on UNIX and Linux, or Administrator on Windows).

6.15.1 Architecture/design

Specifics of the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint:

- Very similar to the Universal Agent:
  - Extensible as described in 6.12, “Custom Resource Model support” on page 368
  - Dynamic discovery of subnodes
  - Single IRA table agent class
  - Data is cached in memory
  - TTL (time-to-live) to expire outdated rows, in order to keep the memory usage to a reasonable size.
- Unlike the Universal Agent:
  - .cat/.atr/ODI files not dynamically generated or uploaded to servers
  - No dynamically generated queries or nodes in the tree (regular seeding needed)
  - Single, custom method of receiving the data (no sockets)

  c. No automatic managed system list creation

d. Stores all received rows of data for warehousing (can be disabled)

e. No CANDLEHOME or any regular utilities (minimal install image)

f. Installed via Tivoli Management Framework

6.15.2 What the monitoring agent is and is not

This section describes what the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint is and what it is not.

- It has two primary roles to perform toward IBM Tivoli Monitoring 6.1:

  a. Displays resource model health status and indications (such as **`wdmlseng`**).

  b. Displays real-time resource model metrics and logs data for historical usage.

- It is focused on data collection; for example, starting and stopping resource models via the agent or TEPS is not possible as it is with the Web Health Console.
- It does not take events of the IBM Tivoli Monitoring engine into account, so if you have response action set for your resource model, you should create the corresponding situations with IBM Tivoli Monitoring 6.1.
- Behaves, mostly, like any other IBM Tivoli Monitoring 6.1 agent.
- Heartbeating.
- Collects data for real-time visualization (TEPS requests for resource model health/indications and resource model views)
- Records the data for warehousing (stored locally or on the TEMS), if enabled

6.15.3 Internal components

Figure 6-34 describes the internal components of the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint.

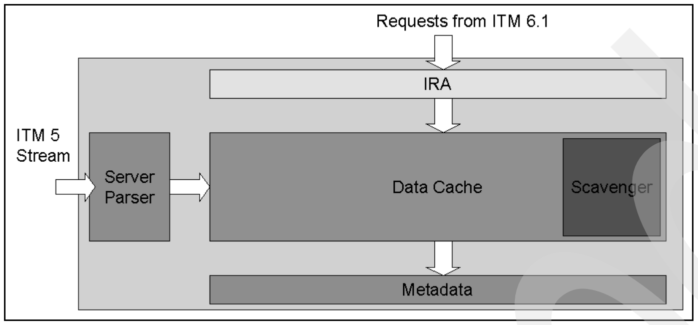


Figure 6-34 Internal Components

- IRA (Intelligent Remote Agent framework)

  APIs and framework for requests, such as exporting data to warehouse, and so on
- Server Parser

  Receives the data from IBM Tivoli Monitoring 5.x and parses the stream into the internal format
- Data Cache

  Due to the IBM Tivoli Monitoring 6.1 minimum historical collection time being five minutes and the resource model cycle time being highly configurable and likely to be around one minute or less, the agent keeps a separate memory buffer containing all received logged data without keying so that the historical situation can retrieve all logged data since it last ran.

  This is done only if a historical situation is active against the table. Stopping the historical situation causes the historical data cache for that table to be discarded.

  After the historical situation retrieves the data, the memory buffer is cleared so that duplicates are not reported and new data samples can be stored. For each table where Historical Data Collection is enabled, a separate binary file will be produced just as with any regular IBM Tivoli Monitoring 6.1 agent.

  To summarize, these are the main roles of the Data Cache:

  – Central piece of the agent.

  – Keeps data in memory and uses keying information to update existing rows in memory.

- Duplicates rows into separate cache if historical collection enabled
- Uses resource model cycle time to calculate time-to-live of rows. TTL is set at 2.5 * cycle for the resource model. Includes all tables, except for the Messages table which defaults to 1 hour TTL. Any data logged before cycle time is available (logged data received before health data), will default to having 1 hour TTL until data is refreshed.
- Scavenger thread (part of the cache code) runs every minute to delete expired rows based on TTL. If data is disappearing before it should, then data is not being refreshed as often as it should.
- Most tracing is not active by default to prevent the log file from growing too big and useless. Most problems are surfaced via the Messages table.

6.15.4 Integration with Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint and IBM Tivoli Monitoring 6.1: How does it work?

Figure 6-35 describes the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint integration with IBM Tivoli Monitoring 6.1.

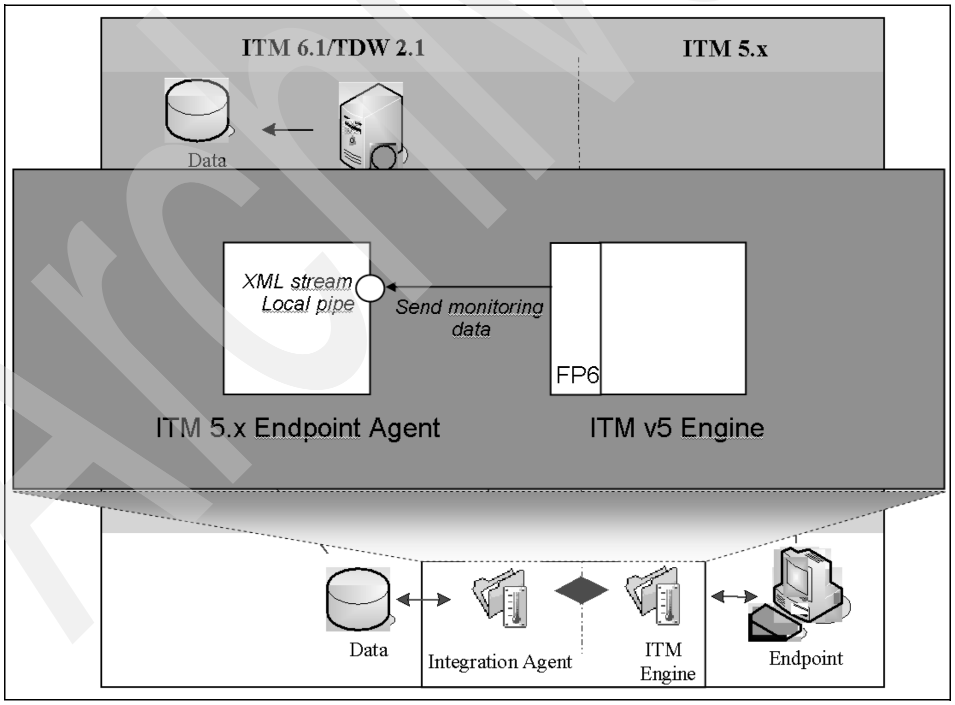


Figure 6-35 Monitoring Agent integration with IBM Tivoli Monitoring 6.1

Agent receives data from IBM Tivoli Monitoring 5.x engine via STDIN (no sockets, local communication only through a local pipe, no special configuration or gateway needed). Data is a subset of XML.

The Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint listens on STDIN file descriptor for data from the IBM Tivoli Monitoring engine. When a defineLogInst is received, the agent will try to match it against a loaded metadata file. If no file matches it, the agent writes a sample metadata file from the data contained in the defineLogInst data under $LCF_DATDIR/LCFNEW/KTM/metadata subdirectory of the agent.
The filename will be: <Category>_<RM>_<Context>_<Resource>.xml.sample.

In the event that the logInstEx does not match a loaded metadata file, the agent will put the data into a generic table, which enables the table to be stored and historically collected until the custom resource model integration procedure has been completed.

Refer to 6.12, "Custom Resource Model support" on page 368 for details about the integration procedure.

Missing columns will be treated as empty values or zero value depending on the column type. Extra unknown columns will also be logged into the generic table to avoid data loss.

Integer numbers can have a scaling factor applied as described in 6.12.7, "Metadata files" on page 382. Enumerated values are supported. Numeric units are not available unless it is part of the attribute name.

IBM Tivoli Monitoring 5.x has been changed to start the ktmcma if one of these conditions is met:

- DataSeeding configured to ITM6 or BOTH:

ITM5	IBM Tivoli Monitoring engine behaves exactly as before and does not start or send any data to the ktmcma agent (default value).
ITM6	IBM Tivoli Monitoring engine does not create the endpoint database and only sends logged data to the ktmcma agent. This reduces the engine memory and CPU consumption and disables the old IBM Tivoli Data Warehouse path.
BOTH	Combination of both flags that can be used in a transition to the ITM 6.1 infrastructure and warehousing.

Note: If the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint is not installed, **`witm61agt`** has not been run against the Endpoint. Use the **`wdmepconfig`** command to set DataSeeding to ITM6 or BOTH. The behavior will be the following:

- **ITM6**: Considering that the ktmcma is not started because the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint is not installed, the IBM Tivoli Monitoring engine will not do anything and will not process either ITM5 or ITM6 data.
- **BOTH**: The IBM Tivoli Monitoring engine will continue to work as if the DataSeeding is set to ITM5.

When this condition is met, the Tmw2k.exe on Windows, or the Java process on UNIX - Linux, starts the ktmcma, redirecting it to the STDIN.

Stopping or recycling the IBM Tivoli Monitoring 5.1.2 engine causes the ktmcma to read the EOF from the IBM Tivoli Monitoring engine and shut down gracefully.

- When the IBM Tivoli Monitoring engine needs to send logging data to the ktmcma for processing, it checks for the existence of the ktmcma process (ktmcma.exe on Windows). If it is not there, it attempts to start the process. If the start procedure is not successful, it retries the operation based on an internal cycle.

6.15.5 XML streams

The engine must know which resource models belong to the same PAC and what the node type name in the metadata file is for that PAC.

Custom resource models that want to take advantage of the Integration Agent linkage are also supported. We use the Category setting in the resource model to determine the node type that the resource model applies to. Using Category enables PAC and customer-created resource models to be grouped into a single metadata (by PAC or customer grouping for custom resource models) without requiring any type of configuration update.

The IBM Tivoli Monitoring engine will be supplying data for two or more different metadata files.

In order to provide a clean separation between IBM Tivoli Monitoring 5.1.2 and the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint, the format of the data being passed from IBM Tivoli Monitoring 5.1.2 will be a unique format to support the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint linkage. The IBM Tivoli Monitoring engine will send data about DefineLoginst, LogInstEx, and overall health data (resource model status, indication instances, and indication metrics).

Refer to 6.12, “Custom Resource Model support” on page 368 for details about the custom resource model integration.

The following is an example of the XML streams used by the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint:

- defineLogInst

  On each defineLogInst statement, the IBM Tivoli Monitoring engine passes this data to the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint.

  ```
  <defineLogInst version="1.0"><category name="Windows"/><rm
  name="TMW_ParamServices"/><context name="Services Status"/><rsc
  name="Service"/><ksAttr name="Service"/><sAttr name="State"/><sAttr
  name="Status"/></defineLogInst>
  ```

  Agent looks up a loaded metadata matching the category, resource model name, context, and resource. If not found, a <category>_<RM>_<context>_<resource>.sample file will be created under the $LCFDATDIR/LCFNEW/KTM/metadata directory so that you can use the collected sample files to produce a metadata file for the seed generator tool as described in 6.12, “Custom Resource Model support” on page 368.

- itmHealth

  The health information provided by the IBM Tivoli Monitoring engine is also passed with this format:

  ```
  <itmHealth version="1.0"><sampletimestamp time="1130577628"/><table
  name="Resource_Model_Health"/><sAttr
  name="ProfileName">WINDOWS_UD#istanbul-region</sAttr><sAttr
  name="RMName">TMW_ParamServices</sAttr><sAttr
  name="Status">Running</sAttr><nAttr name="CycleTime">300</nAttr><nAttr
  name="Health">100</nAttr></itmHealth>
  ```

  Note: sampletimestamp is always expected for both logInstEx and itmHealth tags.

  This XML stream is used to provide resource model health information to the agent, as with the old Web Health Console. The same stream format is used to provide resource model data, indication instance data, and indication metrics (valid table names are Resource_Model_Health, Indication_Instance, and Indication_Metric).

- logInstEx

  On each LogInstEx statement, the IBM Tivoli Monitoring engine passes the following data and in the event that the content does not match a known metadata file, the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint will map the data into a generic table (KTM_Logged_Data) so that the contents are not lost. The sequence of keys, string, and numeric elements is defined by the defineLogInst within the resource model in order to build the metadata file consistently, but does not have to be followed on the logInstEx data flow.

```
<logInstEx version="1.0"><sampletimestamp time="1130577627"/><category
name="Windows"/><rm name="TMW_ParamServices"/><context name="Services
Status"/><rsc name="Service"/><instance name="Service=DB2DAS00"/><sAttr
name="Service">DB2DAS00</sAttr><sAttr name="State">Running</sAttr><sAttr
name="Status">OK</sAttr></logInstEx>
```

This XML stream represents the actual data logged by the resource model and must match one of the existing loaded metadata files. If no match or extra attributes are detected, data will be logged to the generic table, which resembles the endpoint database. Any missing data will assume default values (empty string or 0 for integers).

6.15.6 Warehousing

- As with any other agent, data can be exported to the 24-hour binary file and then to the Warehouse Proxy.
- Data exported is raw data. Aggregation can be done via Summarization and Pruning agent, unlike old IBM Tivoli Enterprise Data Warehouse 1.2 where only summarized data was placed in IBM Tivoli Enterprise Data Warehouse.
- Unlike other agents, by default all received data is exported because of possible mismatches between resource model cycle times and collection interval. This can result in a lot of data being exported, but can be turned off if needed.

For this purpose, agent cache contains two caches: one for real-time views and one for historical collection (if enabled).

Historical Data Collection

The following provides details related to Historical Data Collection:

- Can be used for any attribute group of an IBM Tivoli Monitoring resource model category. For example:

  Product= ITM 5.x-Windows
  Attribute group=AMW_TMW_LogicalDisk_PercentSpace
- Collection interval: 5,15, 30, or 60 minutes

  This might be slower than the resource model collection interval, so graphs will have to set the “sample time” as the axis attribute in order to display each “real” recording time.

- Collection location: “local” or “TEMS”
- Instance data collected by default
- Summarization interval: hourly, daily, weekly, monthly, quarterly, or yearly
- Pruning settings for each summarization interval and for instance data
- Warehouse upload interval: 1 hour, daily, or off

6.15.7 Performance considerations

Performance considerations about the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint include:

- Agent typically takes very little CPU (average of about 0.3% with few OS resource models running). Memory usage is highly dependent on the amount of data being received and how warehousing is being done (collection interval, type of warehousing being done, and amount of data).
- Each unique category received in the LogInstEx stream corresponds to a node in the tree, so a single system can have multiple managed systems associated with it: :TM for the IRA agent itself, :KTM for the health node, and one per active category (:AMX, :AMW, :AMS, and so on).
- Agent behaves like any other agent. Load on the rest of the system depends on user interaction and other system settings, such as whether data is being collected at TEMS, user queries from TEP, and so on.
- Default seeding includes 208 report nodes, 648 queries, and 292 workspaces.

6.15.8 Manually adding SSL

This section describes the procedure to manually configure the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint to work through SSL (ip.spipe).

These are the steps to be performed on the machine where the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint code is installed:

1. Install 32-bit GSKIT or install the IBM Tivoli Monitoring 6.1 OS Agent.
2. Copy the kdebe (or kdebe.dll for Windows) into the $LCF_LIBDIR.
3. Modify KTMENV to include the following variables (or copy from OS agent environment file if already configured to connect to a TEMS or remote TEMS using the IP.SPIPE connection):
   - ICCRTE_DIR=<GSKIT Install dir>
   - KEYFILE_DIR=<path to key files directory from OS agent or copied over>
   - KDEBE_KEYRING_FILE=<path to keyfile.kdb under KEYFILE_DIR>

  – KDEBE_KEYRING_STASH=<path to keyfile.sth under KEYFILE_DIR>
  – KDEBE_KEY_LABEL=IBM_Tivoli_Monitoring_Certificate_Name

4. Modify the following variables in the KTMENV FILE to use the SSL communication protocol:
   – CT_CMSLIST
     This variable looks like:
     CT_CMSLIST=IP.SPIPE:tems;IP.SPIPE
   – KDC_FAMILIES
     This variable looks like:
     KDC_FAMILIES=IP.PIPE use:n IP.SPIPE port:3660 SNA use:n IP use:n

> **Note:** The right format of KDC_FAMILIES should be:
>
> ▶ PROTOCOL use:n if you want to exclude the use of a specific protocol
>
> ▶ PROTOCOL port:# if you want to use that specific protocol
>
> Do not mix the two formats as this could cause incorrect Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint behavior.

5. Restart the IBM Tivoli Monitoring engine.

Example 6-10 shows the steps used to configure the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint to use SSL on an AIX box.

Example 6-10 SSL implementation for Monitoring Agent

```
The SSL configuration described in this example has been performd with the
following steps:

1) Install the OS agent that install the GSKIT as prerequisite
2) Copy the kdebe into the $LCF_LIBDIR

[root@belfast][/opt/Tivoli/lcf_istanbul/dat/1]-> . ./lcf_env.sh
[root@belfast][/opt/Tivoli/lcf_istanbul/bin/aix4-r1/mrt]-> cp
/opt/IBM/ITM/tmaitm6/aix513/lib/kdebe $LCF_LIBDIR
[root@belfast][/opt/Tivoli/lcf_istanbul/bin/aix4-r1/mrt]-> ls -la
$LCF_LIBDIR/kdebe
total 71407
drwxr-xr-x   2 root     system         1024 Oct 21 19:34 .
drwxr-xr-x   3 root     system          512 Sep 22 11:04 ..
-rwxrwxrwx   1 root     system        49807 Oct 21 19:34 kdebe

3) Modify KTMENV to include the following variables:
```

```
ICCRTE_DIR=<GSKIT Install dir>
KEYFILE_DIR=<path to key files directory from OS agent or copied over>
KDEBE_KEYRING_FILE=<path to keyfile.kdb under KEYFILE_DIR>
KDEBE_KEYRING_STASH=<path to keyfile.sth under KEYFILE_DIR>
KDEBE_KEY_LABEL=IBM_Tivoli_Monitoring_Certificate_Name

GSkit encryption key has been set.
Key File directory: /opt/IBM/ITM/keyfiles

The modified KTMENV file will looks like:
*------------------------------------------------------------------*
* Trace Specification -- Do not modify commented examples.         *
*------------------------------------------------------------------*
* No error tracing. KBB_RAS1=-none-
* General error tracing. KBB_RAS1=ERROR
* Intensive error tracing. KBB_RAS1=ERROR (COMP:ktm ALL)
* Maximum error tracing. KBB_RAS1=ERROR (COMP:ktm ALL) (UNIT:kra ALL)
KBB_RAS1=ERROR
LOGSIZE=5
KBB_RAS1_LOG=$CTIRA_LOG_PATH/$(CTIRA_HOSTNAME)_tm_$(SYSUTCSTART)-.log COUNT=04
PRESERVE=1 LIMIT=$LOGSIZE MAXFILES=8 INVENTORY=$CTIRA_LOG_PATH/$(CTIRA_
HOSTNAME)_tm.inv

* CHANGE THE CT_CMLIST TO INCLUDE THE IP.SPIPE

CT_CMSLIST=IP.SPIPE:nice;IP.SPIPE:paris;

* CHANGED THE KDC_FAMILIES TO DISABLE THE IP.PIPE PORT
* THIS TEST WILL ALLOW A SINGLE IP.SPIPE PATH, BUT YOU CAN SPECIFY ALTERNATE
PROTOCOL PATHS

KDC_FAMILIES=IP.PIPE use:n IP.SPIPE port:3660 SNA use:n IP use:n

* If a protocol is not used specify use:n otherwise port:<port_number>

CTIRA_HOSTNAME=belfast-ep
ATTRLIB=/opt/Tivoli/lcf_istanbul/dat/1/LCFNEW/KTM/ATTRLIB
KBB_VARPREFIX=$
CTIRA_HIST_DIR=/opt/Tivoli/lcf_istanbul/dat/1/LCFNEW/KTM/hist
CTIRA_LOG_PATH=/opt/Tivoli/lcf_istanbul/dat/1/LCFNEW/KTM/logs
KTM_META_PATH=/opt/Tivoli/lcf_istanbul/dat/1/LCFNEW/KTM/metadata
CTIRA_SIT_PATH=/opt/Tivoli/lcf_istanbul/dat/1/LCFNEW/KTM
KBB_SIG1=-dumpoff
KDH_SERVICEPOINT=$(SYSUSER).$(CTIRA_HOSTNAME)_tm

* SSL CONFIGURATION

ICCRTE_DIR=/usr/opt/ibm/gskta
KEYFILE_DIR=/opt/IBM/ITM/keyfiles
```

```
KDEBE_KEYRING_FILE=/opt/IBM/ITM/keyfiles/keyfile.kdb
KDEBE_KEYRING_STASH=/opt/IBM/ITM/keyfiles/keyfile.sth
KDEBE_KEY_LABEL=IBMTivoliMonitoringEncryptionKey

4) Restart the IBM Tivoli Monitoring engine to cause the ktmcma to be recycled
and the new KTMENV file read

5) Verify that the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint is
connected using the SSL port, 3660 by default.

netstat -na from TEMS:

  TCP    9.3.5.88:3660          9.3.5.53:47176         ESTABLISHED

where 9.3.5.88 is the ip address of the Monitoring Agent for IBM Tivoli
Monitoring 5.x Endpoint previously configured.

The Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint will see:

[root@belfast][/opt/Tivoli/lcf_istanbul/dat/1/LCFNEW/KTM]-> netstat -na |grep
3660
tcp4       0      0  9.3.5.53.47185        9.3.5.88.3660         ESTABLISHED
```

6.15.9 Know limitations and workarounds

This section describes some limitations of the GA code of Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint.

Some of these have been addressed for post-GA.

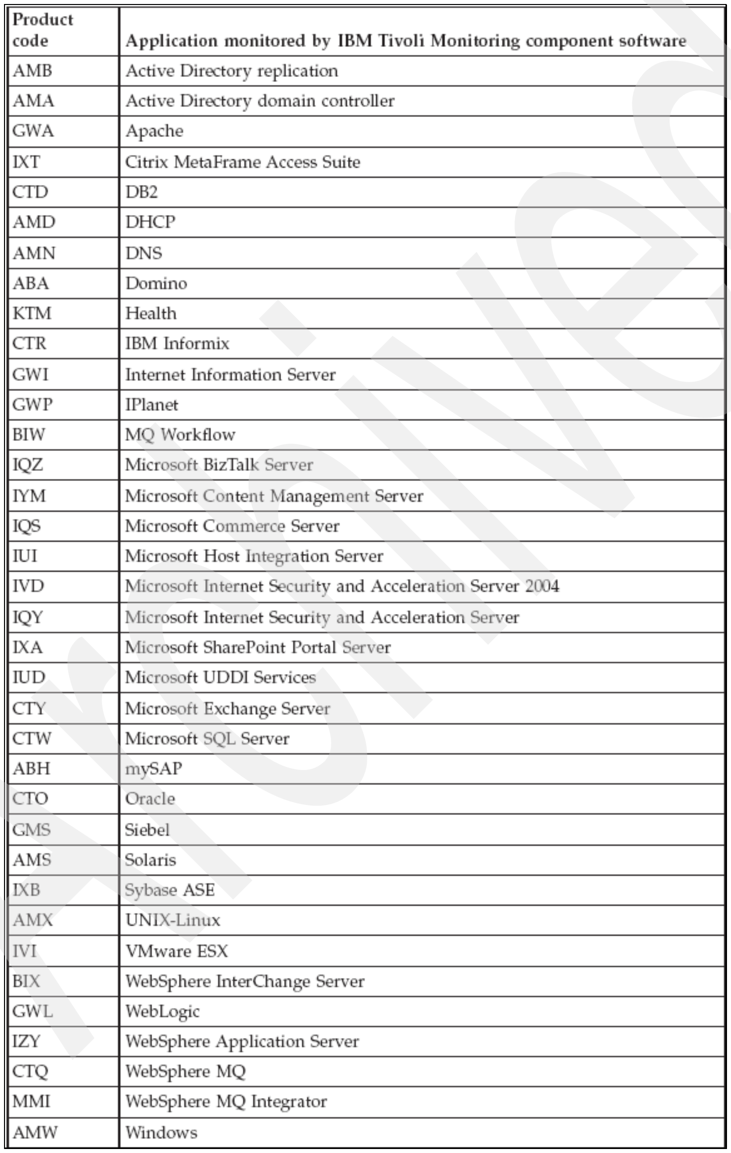
- OS/400 is not supported. (It will be added post-GA.)
- Linux PPC is not supported (not supported by IBM Tivoli Monitoring 6.1).
- No standard IBM Tivoli Monitoring 6.1 utilities due to installation via Framework and to keep installation requirements on the endpoint to a minimal level.
- Manual installation is required for IBM Tivoli Monitoring 6.1 support files. (No upload capability available as in the Universal Agent.)
- No automatic creation of managed system lists as with the Universal Agent, so situations must be distributed to all endpoint subnodes (unless you maintain the managed system lists manually).
- There is no IBM Tivoli Monitoring 6.1 TEP authentication bridge to authenticate TME user and adhere to the defined TME authorization views.
  - To limit access, the TEP logical navigation trees (scopes) for a user have to be defined via the standard TEPS means.

  - No modify rights should be assigned as it disregards these scopes.

- The integration agent cannot be deployed and controlled (start/stop) by means of the IBM Tivoli Monitoring 6.1 infrastructure. This is completely handled by TME and IBM Tivoli Monitoring 5.x mechanisms.
- The ITM6 data collection interval might happen in shorter cycles than the ITM5 data collection interval. ((Graph styles have to be changed to display “Sample Time” instead of “Recording Time.”)

6.16 Product code for IBM Tivoli Monitoring component software

Figure 6-36 lists the product codes that identify IBM Tivoli Monitoring component software. These codes are also called AVA codes. Use these codes when running commands.



Product code	Application monitored by IBM Tivoli Monitoring component software
AMB	Active Directory replication
AMA	Active Directory domain controller
GWA	Apache
IXT	Citrix MetaFrame Access Suite
CTD	DB2
AMD	DHCP
AMN	DNS
ABA	Domino
KTM	Health
CTR	IBM Informix
GWI	Internet Information Server
GWP	IPlanet
BIW	MQ Workflow
IQZ	Microsoft BizTalk Server
IYM	Microsoft Content Management Server
IQS	Microsoft Commerce Server
IUI	Microsoft Host Integration Server
IVD	Microsoft Internet Security and Acceleration Server 2004
IQY	Microsoft Internet Security and Acceleration Server
IXA	Microsoft SharePoint Portal Server
IUD	Microsoft UDDI Services
CTY	Microsoft Exchange Server
CTW	Microsoft SQL Server
ABH	mySAP
CTO	Oracle
GMS	Siebel
AMS	Solaris
IXB	Sybase ASE
AMX	UNIX-Linux
IVI	VMware ESX
BIX	WebSphere InterChange Server
GWL	WebLogic
IZY	WebSphere Application Server
CTQ	WebSphere MQ
MMI	WebSphere MQ Integrator
AMW	Windows

Figure 6-36 Product codes for IBM Tivoli Monitoring component software

7

OMEGAMON XE Upgrade

The IBM Tivoli Monitoring 6.1 infrastructure provides a robust set of tools for monitoring and managing an IT infrastructure and business systems. This chapter provides details about the different steps to upgrade an OMEGAMON XE environment to an IBM Tivoli Monitoring 6.1 environment.

The following topics are described in this chapter:

- Environment
- Upgrade procedure

7.1 Environment

This section describes the software and hardware components used during the OMEGAMON XE upgrade. It also outlines the architecture used to perform the upgrade.

7.1.1 Hardware

A monitoring environment consisting of two Remote CMS systems with multiple agents was used (Table 7-1).

Table 7-1 Hardware configurations

Host name	Hardware Configuration
helsinki	3 GHz CPU 4 GB RAM
florence	3 GHz CPU 1.5 GB RAM
elpaso	375 MHz CPU 512 MB RAM
barcelona	933 MHz CPU 1.1 GB RAM
cairo	3 GHz CPU 4 GB RAM

7.1.2 Operating system and software

Various Windows, Linux, and AIX operating systems were used in the following configuration (Table 7-2).

Table 7-2 Our lab environment for OMEGAMON migration

Function	Host name	Software configuration
Hub CMS	helsinki	**OS:** Windows 2003 **DB2:** 8.2 UDB Workgroup Edition FP10 **OMEGAMON:** CMS V350 CNPS V195.4233a Universal Agent V410 Windows OS Agent V400 DB2 UDB Agent V400

Function	Host name	Software configuration
Remote CMS	florence	**OS:** Windows 2000 SP4 **DB2:** 8.2 UDB Workgroup Edition FP10 **MS SQL Server:** 2000 SP3 **OMEGAMON:** CMS V350 Warehouse Proxy V100 Windows Agent V400 DB2 UDB Agent V400
	elpaso	**OS:** AIX 5.2 **OMEGAMON:** CMS V350 UNIX Agent
Agents only	barcelona	**OS:** RHELv4U1
	cairo	**OS:** Windows 2003 **OMEGAMON**: Windows Agent V400

7.1.3 Lab architecture

The following architecture, described in Figure 7-1, was set up to provide a basic environment to perform all of the functions for prerequisite actions, OMEGAMON XE upgrade, and post-upgrade procedures.

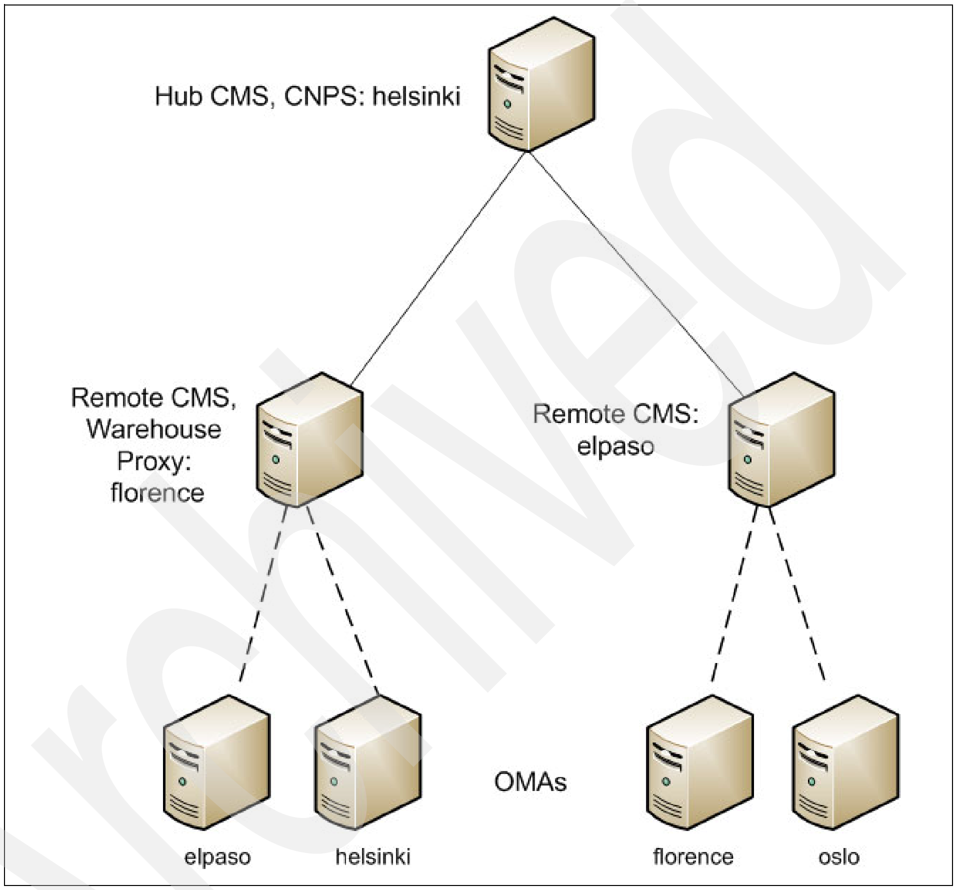


Figure 7-1 OMEGAMON lab architecture

7.2 Upgrade procedure

Upgrading from OMEGAMON XE to IBM Tivoli Monitoring 6.1 is straightforward when carefully planned and implemented. The next sections detail the operations needed to perform a successful upgrade. Table 7-3 introduces the new terminology used in IBM Tivoli Monitoring 6.1.

Table 7-3 OMEGAMON to IBM Tivoli Monitoring 6.1 terminology

OMEGAMON term	IBM Tivoli Monitoring 6.1 term
Candle Management Server® (CMS)	Tivoli Enterprise Monitoring Server (TEMS)
CandleNet Portal® (CNP)	Tivoli Enterprise Portal (TEP)
CandleNet Portal Server (CNPS)	Tivoli Enterprise Portal Server
OMEGAMON Monitoring Agent® (OMA)	Tivoli Enterprise Monitoring Agent (monitoring agent) (TEMA)
OMEGAMON Platform	Tivoli Monitoring Services
Manage Candle Services	Manage Tivoli Enterprise Monitoring Services
Event	Situation event
Seeding	Adding application support

7.2.1 What to do before performing the upgrade

Preliminary information must be gathered to assist the upgrade process. This section details the required information.

Information to gather

1. Verify the current OMEGAMON XE versions and latest release. OMEGAMON XE can only be upgraded to IBM Tivoli Monitoring 6.1 with the 350 and 360 version. Any older versions must first upgrade to OMEGAMON 350 or 360 before upgrading to IBM Tivoli Monitoring 6.1.
2. Confirm that the server platform hardware and operating system comply with the IBM Tivoli Monitoring 6.1 prerequisites. Refer to *IBM Tivoli Monitoring Installation and Setup Guide*, GC32-9407.

   Note: Remember to also verify whether the RDBMS being used in the current environment is supported.

3. Confirm that the user who is performing the upgrade operation has Administrator privileges on the computer (for a Windows system) or has the necessary privileges for the directories, files, and processes that will be running under it.

Tasks to perform

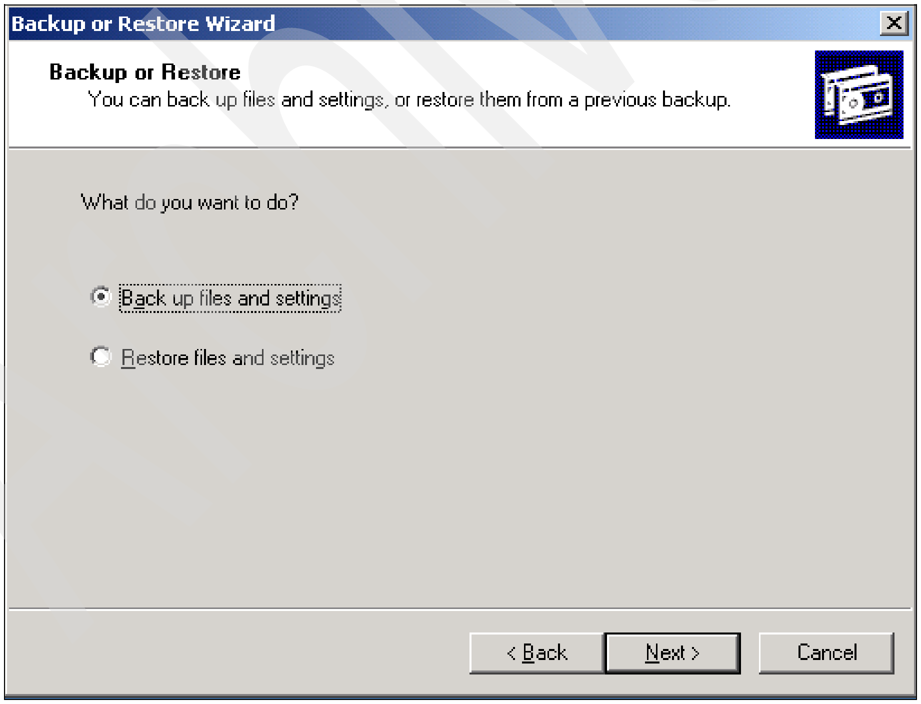
The following tasks must be performed before starting the upgrade process.

System backup

We demonstrate examples of how to take system backups on Windows and UNIX.

To perform a system backup on a Windows server, either use a known backup application or follow this procedure to use the **`ntbackup`** command:

1. From a command prompt, type `ntbackup`.
2. This displays the window shown in Figure 7-2, which presents the option to back up or restore files or directories. Select **Back up files and settings** and click **Next**.


Backup or Restore Wizard

Backup or Restore

You can back up files and settings, or restore them from a previous backup.

What do you want to do?

(●) Back up files and settings

( ) Restore files and settings

Figure 7-2 Windows Backup or Restore initial window

3. From the window in Figure 7-3, you can back up all information by choosing **All information on this computer** or back up specific files or directories (such as c:\candle, c:\Windows and so on) by selecting **Let me choose what to back up**. Click **Next** to continue.

> **Note:** If you choose to back up specific directories and files, include Windows directories because they hold some DLL files that are needed by the CMS process to work correctly.

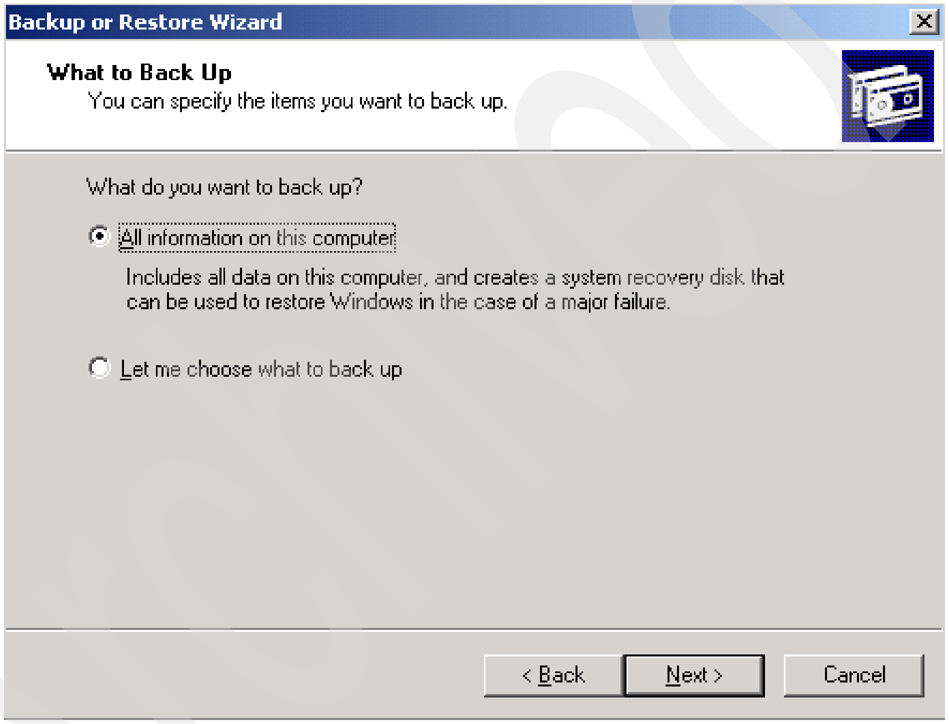
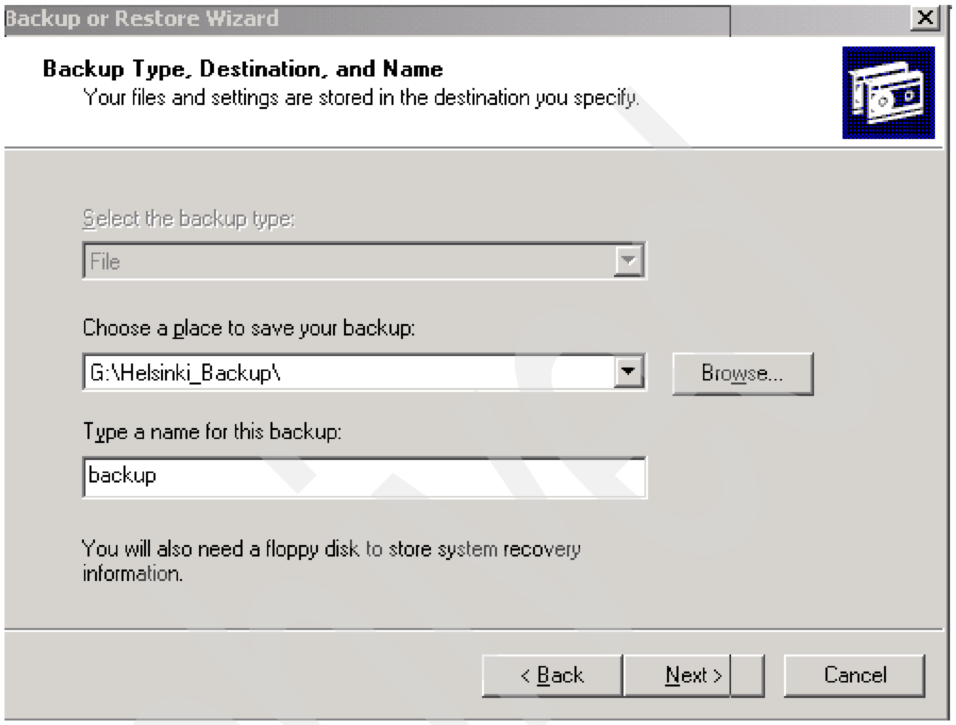

Backup or Restore Wizard
What to Back Up
You can specify the items you want to back up.
What do you want to back up?
(●) All information on this computer
Includes all data on this computer, and creates a system recovery disk that can be used to restore Windows in the case of a major failure.
( ) Let me choose what to back up
< Back
Next >
Cancel

Figure 7-3 What to Back Up

4. In Figure 7-4, choose a place to save your backup and type the name of the backup, then click **Next**.



Backup or Restore Wizard

Backup Type, Destination, and Name

Your files and settings are stored in the destination you specify.

Select the backup type: File

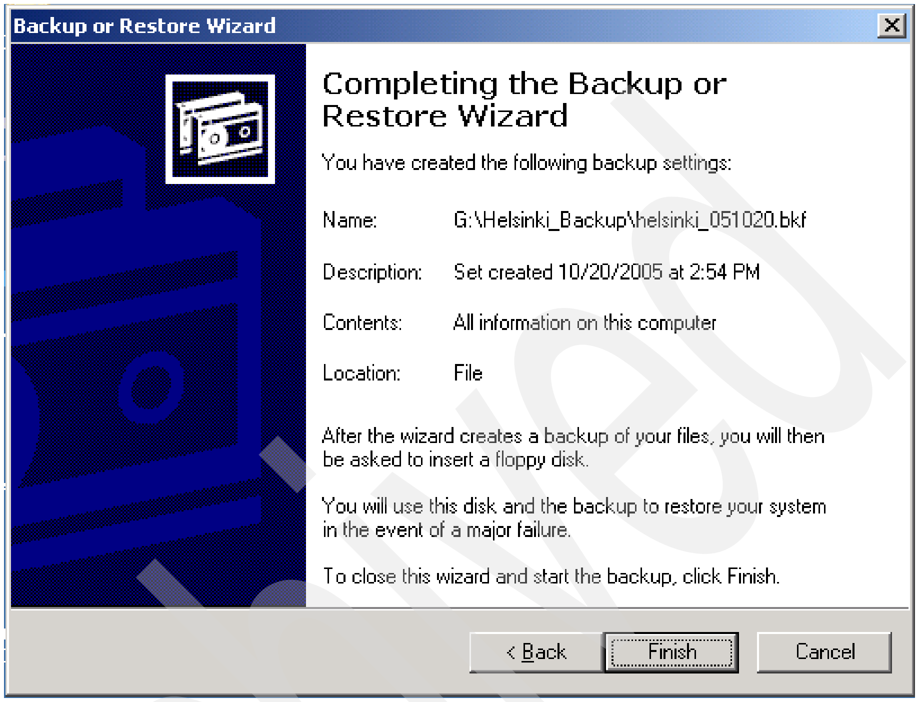
Choose a place to save your backup: G:\Helsinki_Backup\

Type a name for this backup: backup

You will also need a floppy disk to store system recovery information.

Figure 7-4 Specifying the drive and file to back up the system

5. The next window (Figure 7-5) provides the summary of your choices; click **Finish** to start the backup process.


Backup or Restore Wizard

Completing the Backup or Restore Wizard

You have created the following backup settings:

Name: G:\Helsinki_Backup\helsinki_051020.bkf

Description: Set created 10/20/2005 at 2:54 PM

Contents: All information on this computer

Location: File

After the wizard creates a backup of your files, you will then be asked to insert a floppy disk.

You will use this disk and the backup to restore your system in the event of a major failure.

To close this wizard and start the backup, click Finish.

Figure 7-5 Backup final window confirmation

6. When the backup is complete, it asks for a diskette to create a boot diskette. Insert the diskette and press Enter.

The generic backup procedure for UNIX platforms is different.

The easiest way to make a backup of a UNIX environment is to use the **tar** command. Execute the appropriate command to back up all local file systems on a tape drive:

AIX:

```
tar -cvf /dev/rmt0 /
```

Linux:

```
tar -cvf /dev/tap0 /
```

Solaris:

```
tar -cfv /dev/mt0 /
```

CMS database backup

A separate Candle CMS database backup is also recommended. To do so, save all of the following files on a tape, remote drive, CD, or DVD.

Windows:

%CANDLEHOME%\CMS\QA1*.db and QA1*.idx

UNIX:

$CANDLEHOME/CMS/QA1*.db and QA1*.idx

The next procedure should also be done to back up situations and policies.

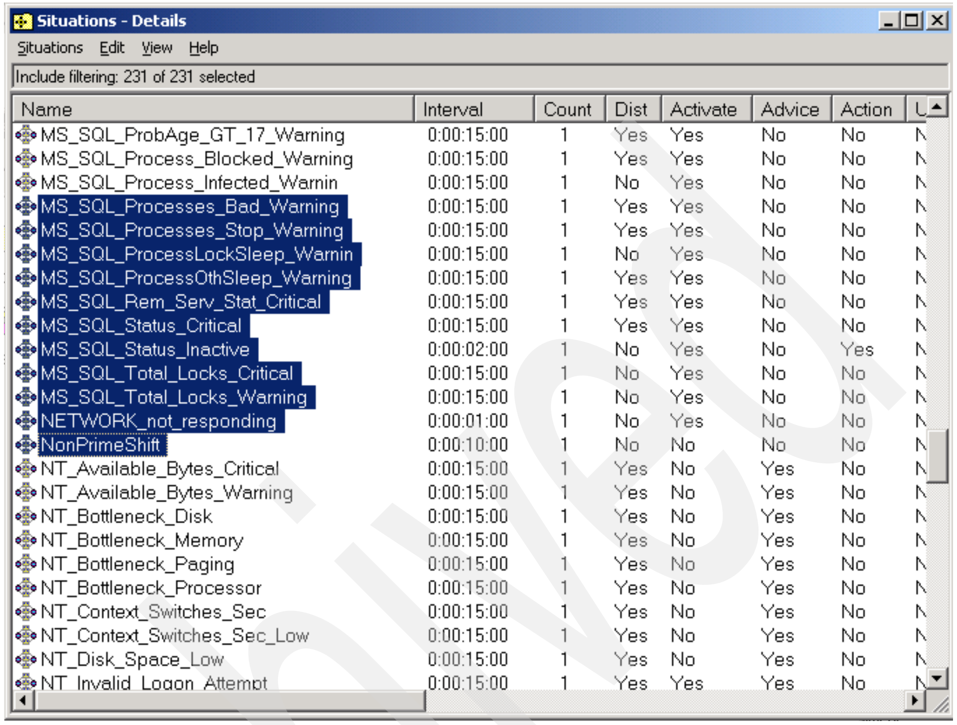
Backing up the CMS data (situations and policies)

An effective way to export CMS data such as situations and policies is provided with the Candle Management Workstation® (CMW).

Use the following procedure to export configured situations:

1. Open the Candle Management Workstation.
2. Double-click **Administration.**
3. Double-click **Situations.**

4. When the list of situations appears, select the situations to export.



Situations - Details

Situations Edit View Help

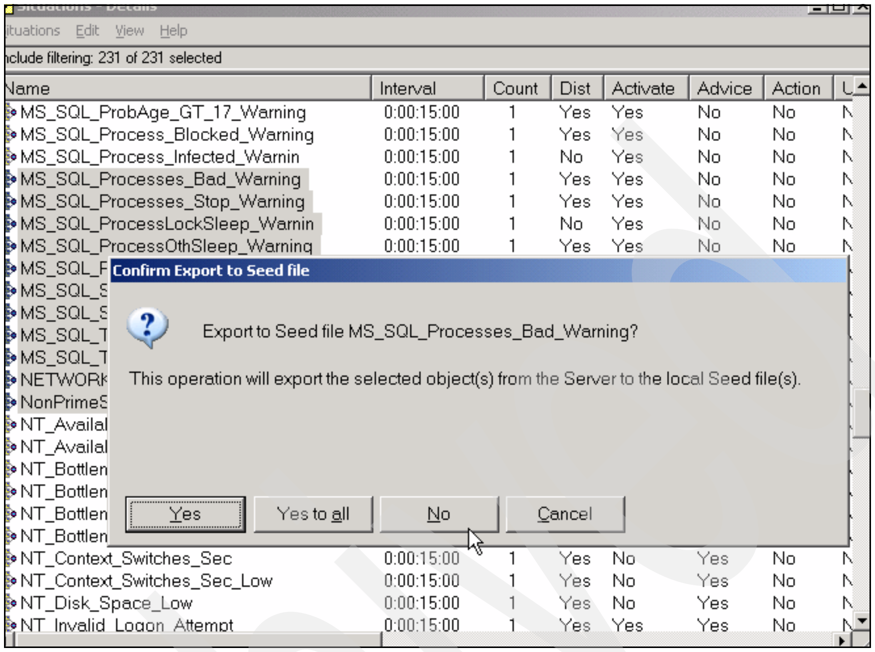
Include filtering: 231 of 231 selected

Name	Interval	Count	Dist	Activate	Advice	Action	U
MS_SQL_ProbAge_GT_17_Warning	0:00:15:00	1	Yes	Yes	No	No	N
MS_SQL_Process_Blocked_Warning	0:00:15:00	1	Yes	Yes	No	No	N
MS_SQL_Process_Infected_Warnin	0:00:15:00	1	No	Yes	No	No	N
MS_SQL_Processes_Bad_Warning	0:00:15:00	1	Yes	Yes	No	No	N
MS_SQL_Processes_Stop_Warning	0:00:15:00	1	Yes	Yes	No	No	N
MS_SQL_ProcessLockSleep_Warnin	0:00:15:00	1	No	Yes	No	No	N
MS_SQL_ProcessOthSleep_Warning	0:00:15:00	1	Yes	Yes	No	No	N
MS_SQL_Rem_Serv_Stat_Critical	0:00:15:00	1	Yes	Yes	No	No	N
MS_SQL_Status_Critical	0:00:15:00	1	Yes	Yes	No	No	N
MS_SQL_Status_Inactive	0:00:02:00	1	No	Yes	No	Yes	N
MS_SQL_Total_Locks_Critical	0:00:15:00	1	No	Yes	No	No	N
MS_SQL_Total_Locks_Warning	0:00:15:00	1	No	Yes	No	No	N
NETWORK_not_responding	0:00:01:00	1	No	Yes	No	No	N
NonPrimeShift	0:00:10:00	1	No	No	No	No	N
NT_Available_Bytes_Critical	0:00:15:00	1	Yes	No	Yes	No	N
NT_Available_Bytes_Warning	0:00:15:00	1	Yes	No	Yes	No	N
NT_Bottleneck_Disk	0:00:15:00	1	Yes	No	Yes	No	N
NT_Bottleneck_Memory	0:00:15:00	1	Yes	No	Yes	No	N
NT_Bottleneck_Paging	0:00:15:00	1	Yes	No	Yes	No	N
NT_Bottleneck_Processor	0:00:15:00	1	Yes	No	Yes	No	N
NT_Context_Switches_Sec	0:00:15:00	1	Yes	No	Yes	No	N
NT_Context_Switches_Sec_Low	0:00:15:00	1	Yes	No	Yes	No	N
NT_Disk_Space_Low	0:00:15:00	1	Yes	No	Yes	No	N
NT_Invalid_Logon_Attempt	0:00:15:00	1	Yes	Yes	Yes	No	N

Figure 7-6 Candle Management Workstation and situation administration

5. Click **Edit** → **Export Seed**.

6. Select **Yes** to all as shown in Figure 7-7.



nclude filtering: 231 of 231 selected

Name	Interval	Count	Dist	Activate	Advice	Action	L
MS_SQL_ProbAge_GT_17_Warning	0:00:15:00	1	Yes	Yes	No	No	N
MS_SQL_Process_Blocked_Warning	0:00:15:00	1	Yes	Yes	No	No	N
MS_SQL_Process_Infected_Warnin	0:00:15:00	1	No	Yes	No	No	N
MS_SQL_Processes_Bad_Warning	0:00:15:00	1	Yes	Yes	No	No	N
MS_SQL_Processes_Stop_Warning	0:00:15:00	1	Yes	Yes	No	No	N
MS_SQL_ProcessLockSleep_Warnin	0:00:15:00	1	No	Yes	No	No	N
MS_SQL_ProcessOthSleep_Warning	0:00:15:00	1	Yes	Yes	No	No	N
MS_SQL_F							
MS_SQL_S							
MS_SQL_S							
MS_SQL_T							
MS_SQL_T							
NETWORK							
NonPrimeS							
NT_Availal							
NT_Availal							
NT_Bottlen							
NT_Bottlen							
NT_Bottlen							
NT_Bottlen							
NT_Context_Switches_Sec	0:00:15:00	1	Yes	No	Yes	No	N
NT_Context_Switches_Sec_Low	0:00:15:00	1	Yes	No	Yes	No	N
NT_Disk_Space_Low	0:00:15:00	1	Yes	No	Yes	No	N
NT_Invalid_Logon_Attempt	0:00:15:00	1	Yes	Yes	Yes	No	N

Confirm Export to Seed file

Export to Seed file MS_SQL_Processes_Bad_Warning?

This operation will export the selected object(s) from the Server to the local Seed file(s).

Figure 7-7 Confirming the situations to be exported

7. Enter the package name and click OK.

The same operation can be performed to export other data, or to import data that had previously been exported.

To back up the whole CMS on one system, back up the complete Candle home directory (for example, c:\candle).

Note: Remember that there is no such thing as a useless backup.

User ID, query, workspace, and navigator configuration backup

Back up the CNPS configuration using the migrate-export.bat tool. This tool migrates all defined tables in the export.params file (such as user ID, queries, workspace, navigators, terminal scripts) using the following command:

```
migrate-export.bat
```

Note:

- The migrate-export.bat facility detects the CNPS database and creates the SQL file script that will be used to restore the CNPS services.
- Importing backup data

  Use the following procedure to restore the CNPS to its initial state:

  – To import the data back into CNPS, copy the presentation files back to the CNPS directory.

  – Copy saveexport.sql file to \cnps\sqllib and run migrate_import.bat.

Stopping and starting Candle components

Before initiating the upgrade procedure, stop all components in the Candle environment.

Always stop the services in this order:

1. CandleNet Portal Server
2. Agents
3. CMS

Tip: Always start the services in the reverse order.

The OMEGAMON XE components can be stopped and started from Manage Candle Services. Use the following steps:

Windows:

1. Right-click the component (such as a specific agent or the Tivoli Enterprise Portal Server) to be stopped or started.
2. Click **Start**, **Stop**, or **Recycle** (Windows only) from the menu; wait for the Status to show `Started` or `Stopped` accordingly.

UNIX:

To start or stop components from the Manage Candle Service window:

1. Highlight the component to start or stop.
2. Do one of the following:
   – Click **Go** (Green) light or **Stop** (Red) light on the menu bar.
   – From the menu bar, select **Actions** → **Start Service** or **Stop Service**.

Important: When upgrading a hub monitoring server, if it is already configured to work with the hot standby feature, stop all agents that connect to the monitoring server.

You also can use the following command to stop the CMS and the agent on a UNIX server:

- To stop the server:

  ```
  CandleServer stop <cms_name>
  ```

  <cms_name> is the name of the CMS, not its host name.

- To stop an agent:

  ```
  CandleAgent stop pc
  ```

  pc is the product code (for example, ux for UNIX Agent, nt for Windows Agent, um for Universal Agent, and so on).

When all components have been stopped and configured to start manually, restart the computer on which IBM Tivoli Monitoring 6.1 is being installed.

Configure the components' startup to be manual

Do the following to configure the components' startup to be manual:

1. Open the Candle Manage Services window.
2. Right-click the component (such as a specific agent or the Tivoli Enterprise Portal Server).
3. Click **Change Startup**.
4. In the Start up box select **Manual** and click **OK**.

Configure the agents

Stop all agent components in the environment.

Important: Do *not* take the following steps in an OMEGAMON migration:

- Do not start upgrading the environment without defining an upgrade plan.
- Do not upgrade the environment without having a full system backup or Candle environment configuration backup.
- Do not remove any source tables or databases when migrating the Warehouse Proxy data; instead, create another database and export the tables to it.

7.2.2 Pre-upgrade procedures

This section provides information regarding some differences between IBM Tivoli Monitoring 6.1 and OMEGAMON XE, as well as some features or functionality that will be lost or replaced when you upgrade your environment.

ODBC and database communication

Configuration of the different components will be required to communicate properly. This includes configuration of the TEPS, Warehouse Proxy agent, and the Summarization and Pruning agent. This communication is needed to insert and retrieve data from the different databases in the TEPS and in the Warehouse Proxy agent. To perform the database access, those components use ODBC and JDBC (Summarization and Pruning agent). The correct configuration of those interfaces or gateways is crucial to the success of the upgrade. Figure 7-8 shows an example of how the different components are connected and configured among themselves.

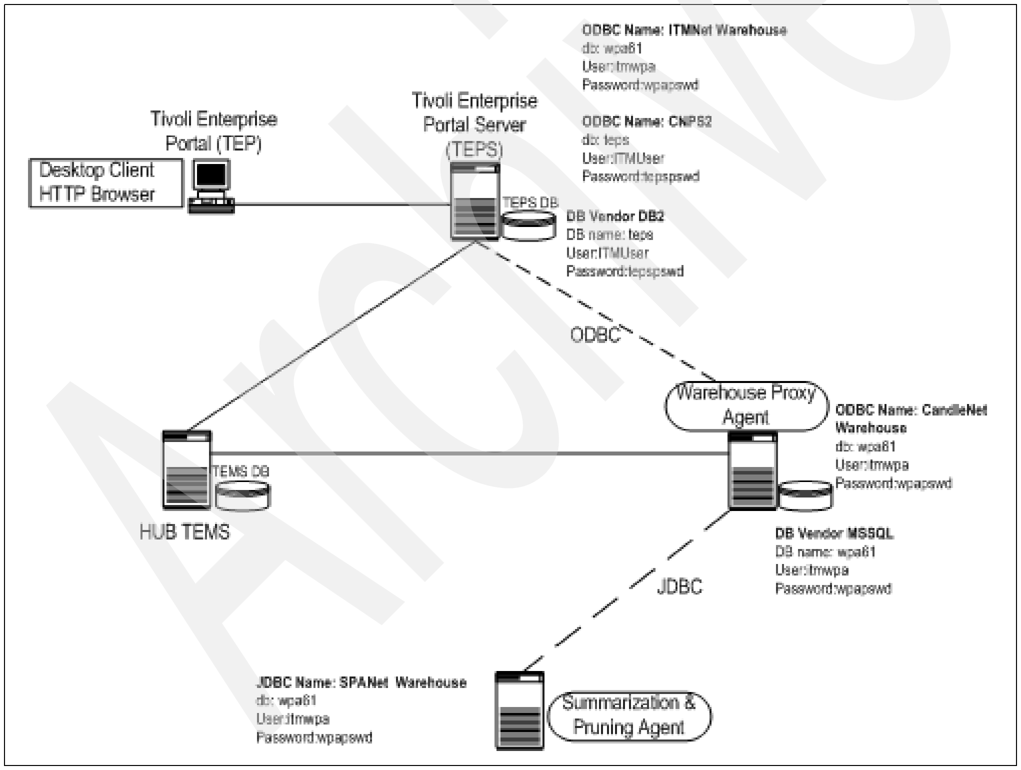


Figure 7-8 Database, ODBC, and JDBC configuration and communication

7.2.3 Installation directory for upgraded components

When upgrading an existing OMEGAMON component to the IBM Tivoli Monitoring level, the installation process installs all new files in the *existing* installation directory; existing files are overwritten. Remember that any documentation references to a default installation directory for IBM Tivoli Monitoring (C:\IBM\ITM on Windows and /opt/IBM/ITM on Linux and UNIX) only apply to new installations, not environments upgraded from OMEGAMON XE.

If fixes or patches have been applied to the OMEGAMON product component, the fixes and patches that were available when IBM Tivoli Monitoring was released are included in the upgraded version.

Configuration settings for upgraded agents

When OMEGAMON V350 or 360 agents are upgraded to IBM Tivoli Monitoring, agent-specific configuration settings regarding the connection to the monitoring server are not maintained. Instead, the IBM Tivoli Monitoring installation uses the default settings for all agents. To ensure that the upgraded agents can connect to the monitoring server immediately, ensure that the default settings for all agents point to the new monitoring server prior to upgrading. To change the default settings for all agents, right-click an agent in Manage Candle Services and click **Set defaults for all agents**.

Candle Management Workstation coexistence

If Candle Management Workstation is used in the OMEGAMON monitoring environment, the installed Candle Management Workstation continues to function after the migration, although it is not officially part of IBM Tivoli Monitoring and no new functionality will be added.

If the OMEGAMON XE for CICS® 3.1.0 product is used, Candle Management Workstation must still be used to configure workloads for Service Level Analysis. After configuring the workloads, use the Tivoli Enterprise Portal client for all other tasks. If Candle Management Workstation is not currently installed (for example, if OMEGAMON XE for CICS on z/OS 3.1.0 is being installed for the first time into an IBM Tivoli Monitoring environment), the Candle Management Workstation that is shipped with the OMEGAMON XE for CICS on z/OS 3.1.0 product must first be installed.

If the Candle Management Workstation is needed, do not install it on the same computer as the Tivoli Enterprise Portal because the Candle Management Workstation will attempt to uninstall the Tivoli Enterprise Portal.

Additional unsupported OMEGAMON functions

The functions in Table 7-4 are no longer supported as part of IBM Tivoli Monitoring.

Table 7-4 Unsupported OMEGAMON functions

Function	**IBM Tivoli Monitoring equivalent function**
CandleClone	Agent deployment as described in 3.2.8, “Tivoli Enterprise Monitoring Agent” on page 95.
CandleRemote	
Event Emitter	Event forwarding using the Tivoli Enterprise Console event synchronization, as described in 3.2.14, “Event synchronization installation” on page 147.
Event Adapter	
GUI installation wizard on UNIX	Use the command line installation option, as described in “Deploying TEMA on a Linux server (using local images)” on page 95.
Silent installation on UNIX using the multi-platform installation program	Silent installation using custom response files, as described in 2.4.3, “Operating system image deployment” on page 54.

CandleNet Portal database

If a DB2 database is used for the CandleNet Portal Server database, the database is converted to UTF-8 format during the upgrade. This might take several minutes, depending on the size of the database. If the default database password “cnps” was used when the CandleNet Portal database was created, a prompt for a new password will appear; this is to comply with the more stringent security provisions in IBM Tivoli Monitoring.

Required Java JRE

The older CandleNet Portal required the Sun™ Java™ JRE; however Tivoli Enterprise Portal in 6.1 requires the IBM Java JRE 1.4.2. This does not have to be installed manually prior to the upgrade: it is installed automatically when choosing to install an IBM Tivoli Monitoring component that requires it.

Migrated information when upgrading from a previous version

If the software is being installed over a previous release (into the same IBM directory) the following information is migrated into the new version:

- On Windows:
  - Port number and communication protocol settings
  - Situations
- On UNIX:
  - UNIX: Situations

7.2.4 Upgrading the different components

This section describes the steps for upgrading an OMEGAMON XE environment. Follow this order:

1. Upgrade CMS, CNP, CNPS, and CMAs (in the order listed).
2. Install the Summarization and Pruning agent.
3. Upgrade/migrate Warehouse Proxy.

Component products

If any of the following products will be installed on the same computer as monitoring agents, they must be installed before the agent is installed:

- Hub Tivoli Enterprise Monitoring Server
- Remote monitoring server (if necessary)
- Tivoli Enterprise Management Agent Framework
- Tivoli Enterprise Portal Server
- Tivoli Enterprise Portal desktop client

In addition, these products must be installed on at least one computer before the agent can be configured properly.

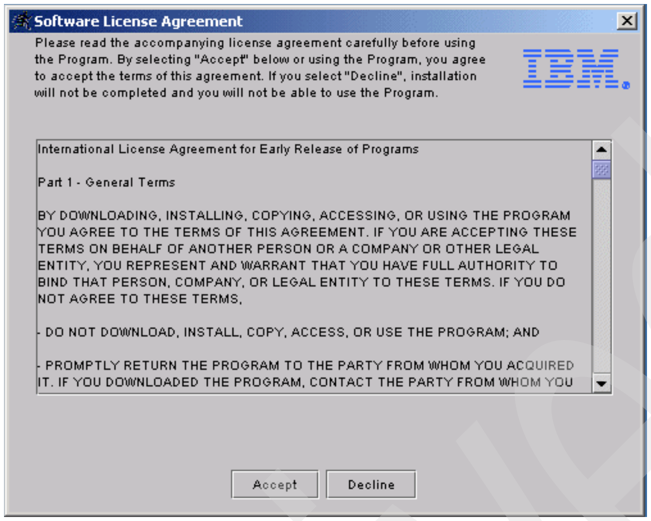
Upgrading a Windows CMS server

Use the following steps to upgrade a hub or a remote CMS server on a Windows computer:

1. Launch the installation wizard by double-clicking the setup.exe file on the installation media.
2. Click **Next** on the welcome window.

Note: When running Windows 2003 or Windows XP with security set to check the software publisher of applications, an error might occur stating that the setup.exe file is from an unknown publisher. Click **Run** to disregard this error message and continue.

3. Click **Accept** to accept the license agreement, as shown in Figure 7-9.



Please read the accompanying license agreement carefully before using the Program. By selecting "Accept" below or using the Program, you agree to accept the terms of this agreement. If you select "Decline", installation will not be completed and you will not be able to use the Program.

International License Agreement for Early Release of Programs

Part 1 - General Terms

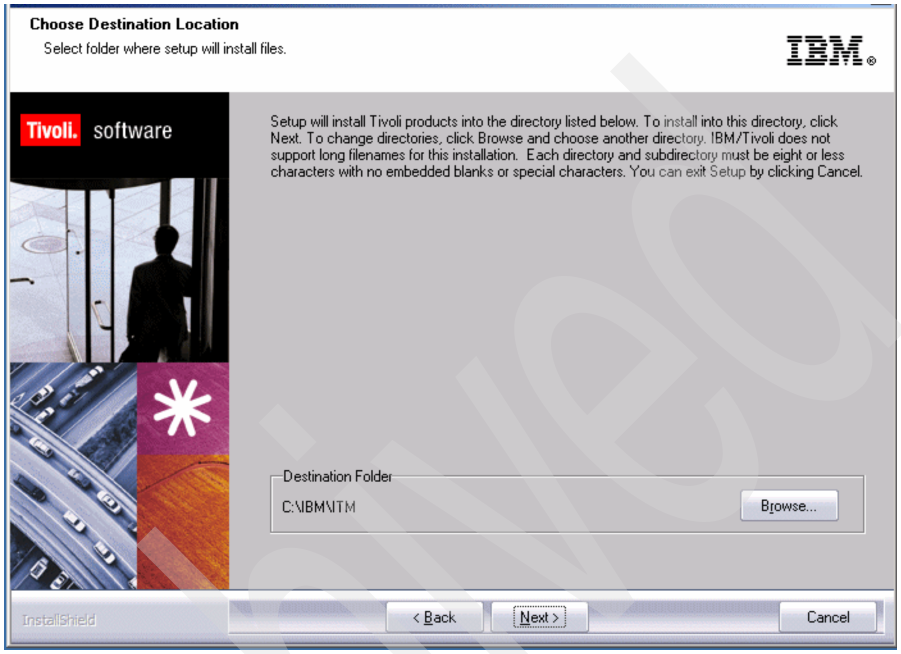
BY DOWNLOADING, INSTALLING, COPYING, ACCESSING, OR USING THE PROGRAM YOU AGREE TO THE TERMS OF THIS AGREEMENT. IF YOU ARE ACCEPTING THESE TERMS ON BEHALF OF ANOTHER PERSON OR A COMPANY OR OTHER LEGAL ENTITY, YOU REPRESENT AND WARRANT THAT YOU HAVE FULL AUTHORITY TO BIND THAT PERSON, COMPANY, OR LEGAL ENTITY TO THESE TERMS. IF YOU DO NOT AGREE TO THESE TERMS,

- DO NOT DOWNLOAD, INSTALL, COPY, ACCESS, OR USE THE PROGRAM; AND

- PROMPTLY RETURN THE PROGRAM TO THE PARTY FROM WHOM YOU ACQUIRED IT. IF YOU DOWNLOADED THE PROGRAM, CONTACT THE PARTY FROM WHOM YOU

Figure 7-9 License agreement window

4. Choose the desired product installation directory (Figure 7-10). The default directory is C:\candle; change it to the current $CANDLEHOME directory if different. Click **Next**.


Choose Destination Location

Select folder where setup will install files.

Setup will install Tivoli products into the directory listed below. To install into this directory, click Next. To change directories, click Browse and choose another directory. IBM/Tivoli does not support long filenames for this installation. Each directory and subdirectory must be eight or less characters with no embedded blanks or special characters. You can exit Setup by clicking Cancel.

Destination Folder: C:\IBM\ITM

Figure 7-10 Installation window

5. Type a 32-bit encryption key. (You can use the default key.) Click **Next** and then click **OK** to confirm the encryption key.

Notes:

- This encryption key is used to established a secure connection (using SSL protocol) between the Hub TEMS and the other components of the IBM Tivoli Monitoring 6.1 environment. Do not use any of the following characters in a key:

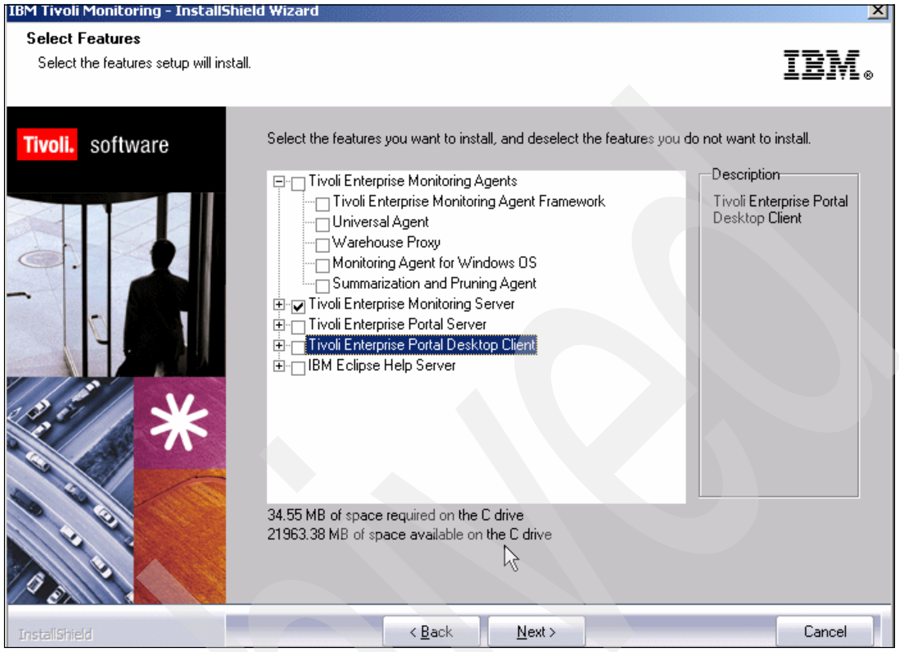
  =

  ,

  |

- Ensure that the chosen key value is documented for use later with other components. Use this key during the installation of any components that communicate with this monitoring server.

6. Select the desired installation components and click **Next**. Figure 7-11 shows the components used for this sample environment.


IBM Tivoli Monitoring - InstallShield Wizard

Select Features

Select the features setup will install.

Select the features you want to install, and deselect the features you do not want to install.

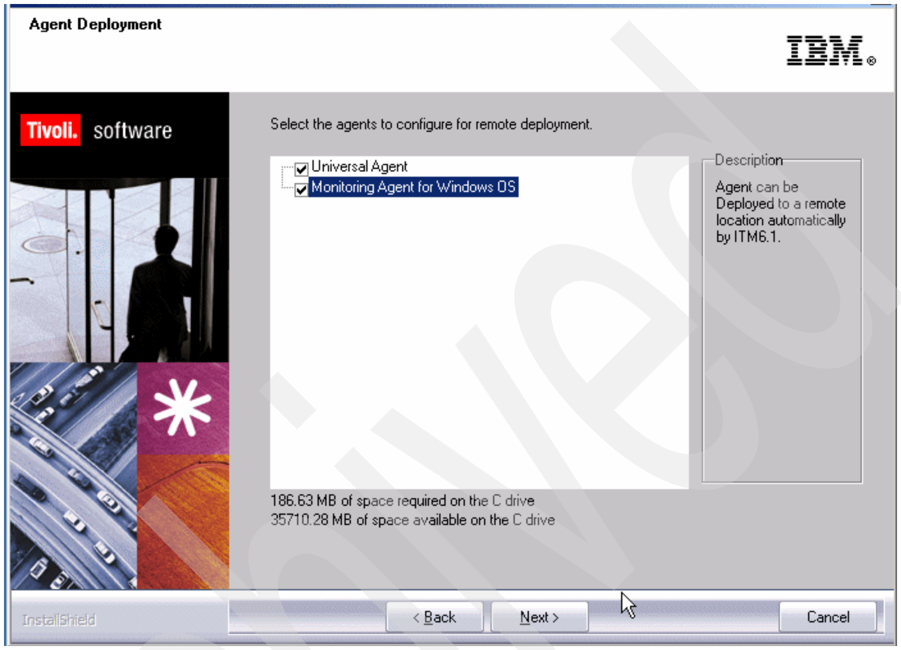
- [ ] Tivoli Enterprise Monitoring Agents
  - [ ] Tivoli Enterprise Monitoring Agent Framework
  - [ ] Universal Agent
  - [ ] Warehouse Proxy
  - [ ] Monitoring Agent for Windows OS
  - [ ] Summarization and Pruning Agent
- [x] Tivoli Enterprise Monitoring Server
- [ ] Tivoli Enterprise Portal Server
- [ ] Tivoli Enterprise Portal Desktop Client
- [ ] IBM Eclipse Help Server

Description: Tivoli Enterprise Portal Desktop Client

34.55 MB of space required on the C drive
21963.38 MB of space available on the C drive

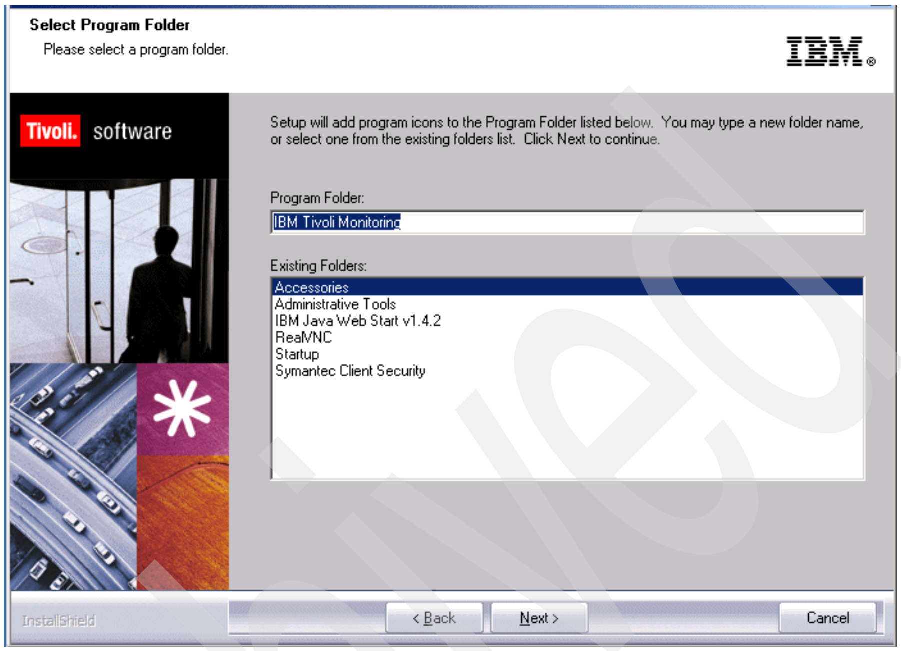
Figure 7-11 List of selected components to be installed

7. If remote deployment of agent software will be desired for later, select those agents that will be deployed, as shown in Figure 7-12. This step creates and populates the deployment depot, from which agents can be deployed at a later time. Click **Next**.


Agent Deployment
IBM.
Tivoli. software
Select the agents to configure for remote deployment.
Universal Agent
Monitoring Agent for Windows OS
Description
Agent can be Deployed to a remote location automatically by ITM6.1.
186.63 MB of space required on the C drive
35710.28 MB of space available on the C drive
InstallShield
< Back
Next >
Cancel
Figure 7-12 Agent list for remote deployment

Note: By default, the depot is located in the <itm_installdir>/CMS/depot directory on Windows and <itm_installdir>/tables/<ms_name>/depot directory on Linux and UNIX. To use a different directory, change the DEPOTHOME value in the kbb.env file.

8. Select a program folder as outlined in Figure 7-13 and click **Next**. The default program folder name is `IBM Tivoli Monitoring`.


Select Program Folder

Please select a program folder.

Setup will add program icons to the Program Folder listed below. You may type a new folder name, or select one from the existing folders list. Click Next to continue.

Program Folder:

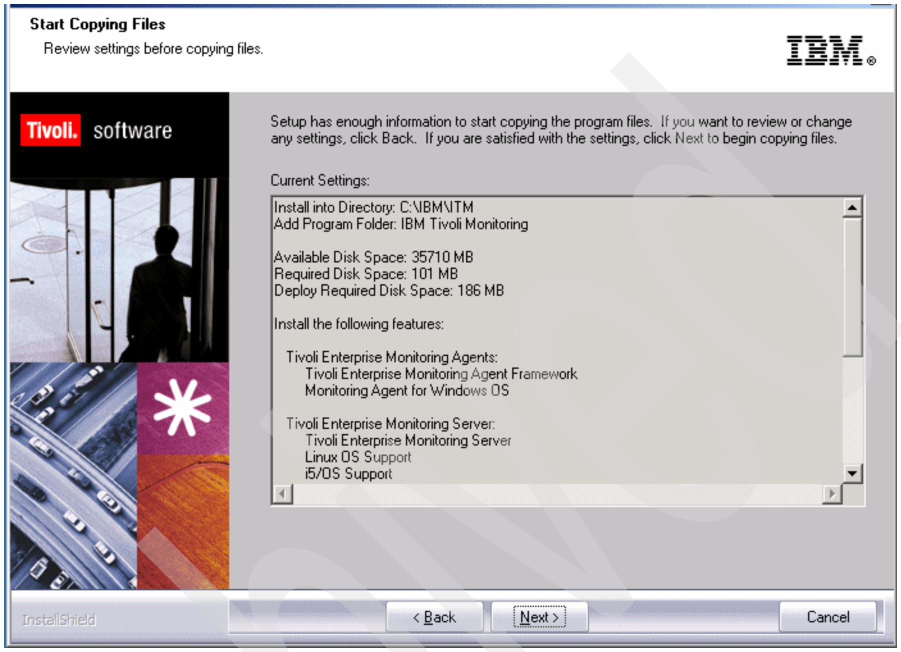
IBM Tivoli Monitoring

Existing Folders:

Accessories
Administrative Tools
IBM Java Web Start v1.4.2
RealVNC
Startup
Symantec Client Security

Figure 7-13 Program Folder for the IBM Tivoli Monitoring 6.1 installation

9. Review the installation summary details. This summary identifies what is being installed and to what paths. Click **Next** to begin the installation of components (Figure 7-14).



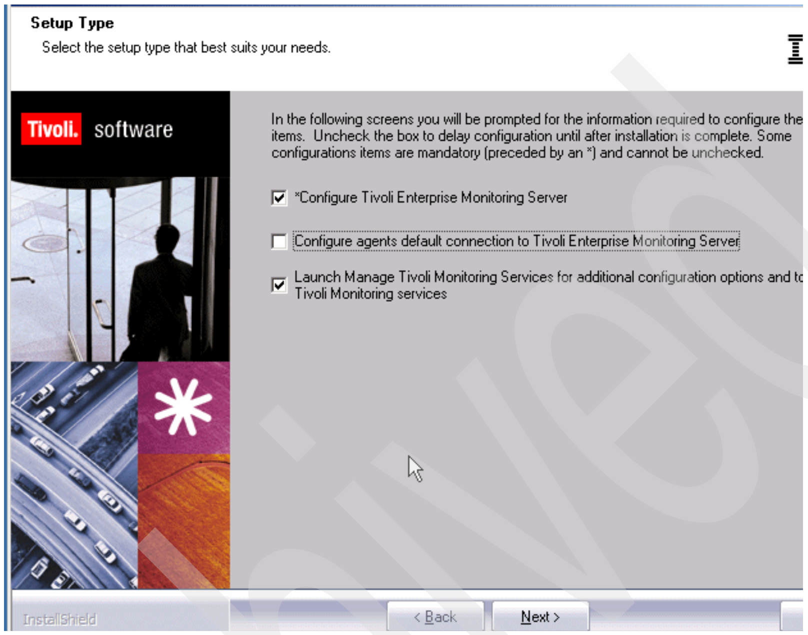
Setup has enough information to start copying the program files. If you want to review or change any settings, click Back. If you are satisfied with the settings, click Next to begin copying files.

Current Settings:

```
Install into Directory: C:\IBM\ITM
Add Program Folder: IBM Tivoli Monitoring

Available Disk Space: 35710 MB
Required Disk Space: 101 MB
Deploy Required Disk Space: 186 MB

Install the following features:

  Tivoli Enterprise Monitoring Agents:
     Tivoli Enterprise Monitoring Agent Framework
     Monitoring Agent for Windows OS

  Tivoli Enterprise Monitoring Server:
     Tivoli Enterprise Monitoring Server
     Linux OS Support
     i5/OS Support
```

Figure 7-14 Installation summary details

10.After the components are installed, a configuration window (Figure 7-15) is displayed with the list of components that can be configured. Select the desired components to configure and click **Next**.


Setup Type

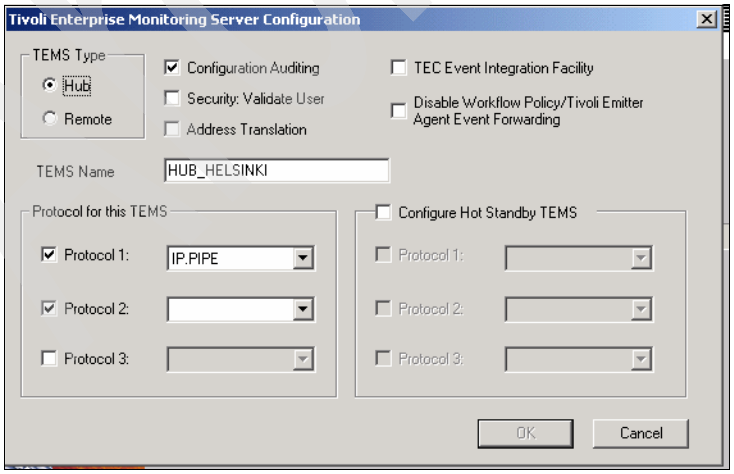
Select the setup type that best suits your needs.

In the following screens you will be prompted for the information required to configure the items. Uncheck the box to delay configuration until after installation is complete. Some configurations items are mandatory (preceded by an *) and cannot be unchecked.

- [✓] *Configure Tivoli Enterprise Monitoring Server
- [ ] Configure agents default connection to Tivoli Enterprise Monitoring Server
- [✓] Launch Manage Tivoli Monitoring Services for additional configuration options and to Tivoli Monitoring services

Figure 7-15 List of components that will be configured

11.Figure 7-16 shows the different options of monitoring server type that can be selected. Select **Hub**.


Tivoli Enterprise Monitoring Server Configuration

TEMS Type: [✓ Hub] Remote

- [✓] Configuration Auditing
- [ ] Security: Validate User
- [ ] Address Translation
- [ ] TEC Event Integration Facility
- [ ] Disable Workflow Policy/Tivoli Emitter Agent Event Forwarding

TEMS Name: HUB_HELSINKI

Protocol for this TEMS
- [✓] Protocol 1: IP.PIPE
- [✓] Protocol 2:
- [ ] Protocol 3:

[ ] Configure Hot Standby TEMS
- [ ] Protocol 1:
- [ ] Protocol 2:
- [ ] Protocol 3:

Figure 7-16 Monitoring server configuration window

12. Verify that the name of this monitoring server is correct in the TEMS field. If it is not, change it. The default name is hub_<hostname>. In the example environment used here, the name HUB_HELSINKI was chosen as the TEMS name.

13. Identify the communications protocol for the monitoring server. There are four options: IP.UDP, IP.PIPE, IP.SPIPE, or SNA. Up to three methods for communication can be selected to allow setup of backup communication methods. If the method identified as Protocol 1 fails, Protocol 2 is used. In the sample environment used here, IP.PIPE was selected as the primary protocol and IP.UDP as the secondary one.

    Note: IP.PIPE protocol uses TCP, so permanent connection is established between the TEMS and the remote servers. This could have an impact on the server performance, because of the number of RPCs that it needs to handle. If using UDP will not cause security or firewall problems in the environment, you are recommended to set up the first protocol as IP.UDP; otherwise use IP.PIPE. Note that if a firewall exists between the TEMS and the agents, IP.UDP probably cannot be used as UDP is commonly blocked.

    Table 7-5 describes the communication protocols that can be used and their definitions. This information is valid for all components in the IBM Tivoli Monitoring 6.1 environment. Only the Hub and Remote TEMS are outlined in this table.

Table 7-5 Communications protocol description

Field	**Description**
IP.UDP Settings: Primary Hub TEMS	
Hostname or IP Address	The host name or IP address for the hub monitoring server.
Port # and/or Port Pools	The listening port for the hub monitoring server.
IP.PIPE Settings: Primary Hub TEMS	
Hostname or IP Address	The host name or IP address for the hub monitoring server.
Port Number	The listening port for the monitoring server. The default value is 1918.

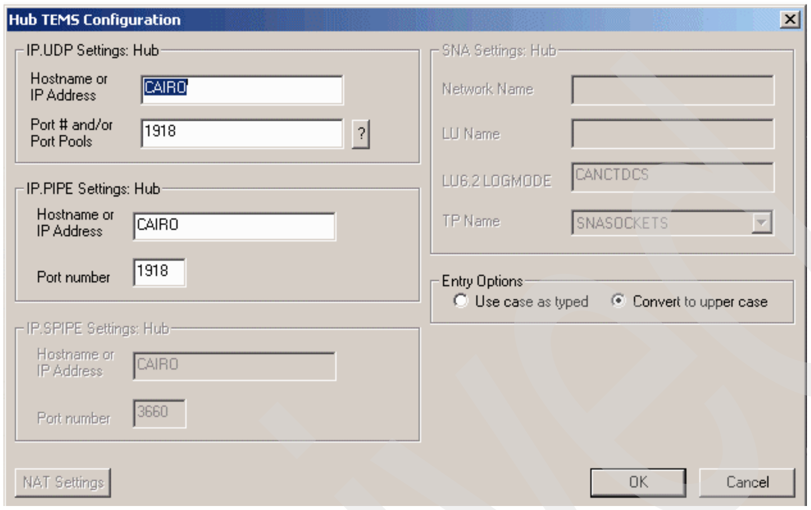
Field	Description
IP.SPIPE Settings: Primary Hub TEMS	
Hostname or IP Address	The host name or IP address for the hub monitoring server.
Port Number	The listening port for the monitoring server. The default value is 3660.
SNA Settings: Remote TEMS	
Local LU Alias	The LU alias
TP Name	The transaction program name for this monitoring server.
SNA Settings: Primary Hub TEMS	
Network Name	The SNA network identifier for your location.
LU Name	The LU name for the monitoring server. This LU name corresponds to the Local LU Alias in your SNA communications software.
LU 6.2 LOGMODE	The name of the LU6.2 LOGMODE. The default value is .CANCTDCS.
TP Name	The transaction program name for the monitoring server.

14. To forward situation events to IBM Tivoli Enterprise Console, select the **TEC Event Integration Facility** option.

    The Configure Hot Standby TEMS option was not selected in the sample environment, because it is easier to separately set up later after all TEMSs are installed and properly configured. The Disable Workflow Policy/TivoliEmitter Agent Event Forwarding option was not selected either.

    Click **OK**.

15. The next window gives options to configure the hub server name or IP address and communication port that the communication protocol will use. Complete the fields as shown in Figure 7-17.



Hub TEMS Configuration

IP.UDP Settings: Hub
- Hostname or IP Address: CAIRO
- Port # and/or Port Pools: 1918

IP.PIPE Settings: Hub
- Hostname or IP Address: CAIRO
- Port number: 1918

IP.SPIPE Settings: Hub
- Hostname or IP Address: CAIRO
- Port number: 3660

SNA Settings: Hub
- Network Name:
- LU Name:
- LU6.2 LOGMODE: CANCTDCS
- TP Name: SNASOCKETS

Entry Options
- Use case as typed [✓ Convert to upper case]

NAT Settings | OK | Cancel

Figure 7-17 Host and communication protocol configuration window

16. If it is certain that the values for all of these fields are typed with exactly the correct cases (upper and lower cases), then the **Use case as typed** option can be chosen. However, because IBM Tivoli Monitoring is case-sensitive, consider selecting **Convert to upper case** to reduce the chance of user error. Click **OK** to continue.

17. The next configuration step is to add application support to the monitoring server, such as the workspaces and situations for agents.

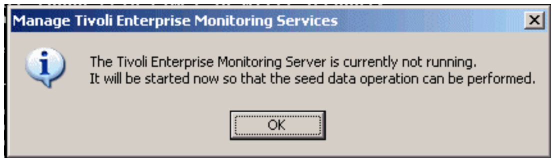
18. After the configuration is complete, a prompt appears asking to seed the TEMS. Specify the location of the monitoring server. There are two options:

   – On this computer

   – On a different computer

   Chose the first option: On **this computer**, and click **OK**.

19. Because the monitoring server is not currently running, it will start automatically before the process begins. Click **OK** on the window in Figure 7-18.


Manage Tivoli Enterprise Monitoring Services

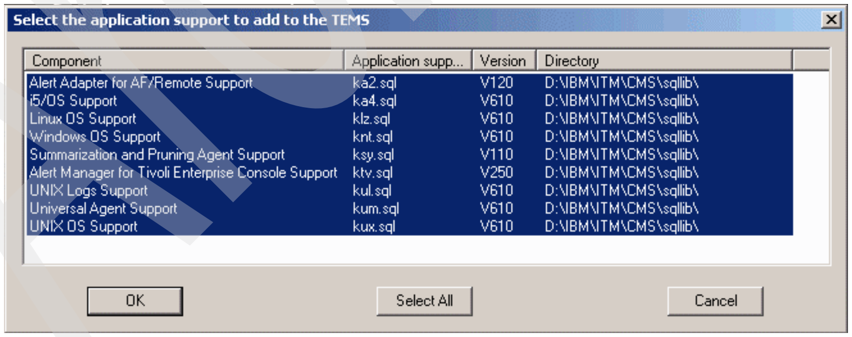
The Tivoli Enterprise Monitoring Server is currently not running.
It will be started now so that the seed data operation can be performed.

Figure 7-18 Monitoring server start confirmation window

20. Select the application support items to add to the monitoring server. By default, all available application support is selected; you are recommended to leave all the components selected, so that they can all be seeded. Click **OK**.

Note: Seeding adds product-specific data from the monitored resources to the monitoring server. During this process, fields are created in the TEMS proprietary database for the agents that were chosen. This basically enables the TEMS to work with the data from these agents. The same goes for the TEPS, except that the necessary tables are created in the TEPS RDBMS.

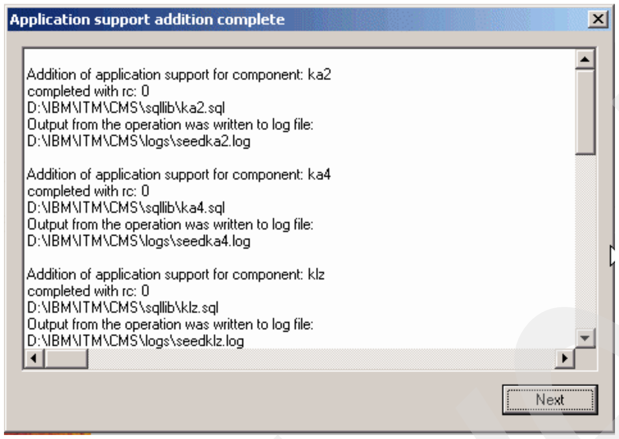
If the seed data is for an agent that reports to a remote monitoring server, complete this process for both the hub and the remote monitoring server. A hub monitoring server should be running before you proceed with a remote monitoring server seed.


Select the application support to add to the TEMS

Component	Application supp...	Version	Directory
Alert Adapter for AF/Remote Support	ka2.sql	V120	D:\IBM\ITM\CMS\sqllib\
i5/OS Support	ka4.sql	V610	D:\IBM\ITM\CMS\sqllib\
Linux OS Support	klz.sql	V610	D:\IBM\ITM\CMS\sqllib\
Windows OS Support	knt.sql	V610	D:\IBM\ITM\CMS\sqllib\
Summarization and Pruning Agent Support	ksy.sql	V110	D:\IBM\ITM\CMS\sqllib\
Alert Manager for Tivoli Enterprise Console Support	ktv.sql	V250	D:\IBM\ITM\CMS\sqllib\
UNIX Logs Support	kul.sql	V610	D:\IBM\ITM\CMS\sqllib\
Universal Agent Support	kum.sql	V610	D:\IBM\ITM\CMS\sqllib\
UNIX OS Support	kux.sql	V610	D:\IBM\ITM\CMS\sqllib\

Figure 7-19 Application support to be added to TEMS

21. Verify that each added application support for the components has a return code (rc) equal to 0, as shown in Figure 7-20, and click **Next**.



```
Addition of application support for component: ka2
completed with rc: 0
D:\IBM\ITM\CMS\sqllib\ka2.sql
Output from the operation was written to log file:
D:\IBM\ITM\CMS\logs\seedka2.log

Addition of application support for component: ka4
completed with rc: 0
D:\IBM\ITM\CMS\sqllib\ka4.sql
Output from the operation was written to log file:
D:\IBM\ITM\CMS\logs\seedka4.log

Addition of application support for component: klz
completed with rc: 0
D:\IBM\ITM\CMS\sqllib\klz.sql
Output from the operation was written to log file:
D:\IBM\ITM\CMS\logs\seedklz.log
```

Figure 7-20 Application addition support window

22. The next configuration step (Figure 7-21) configures the default communication between any IBM Tivoli Monitoring component and the hub monitoring server.

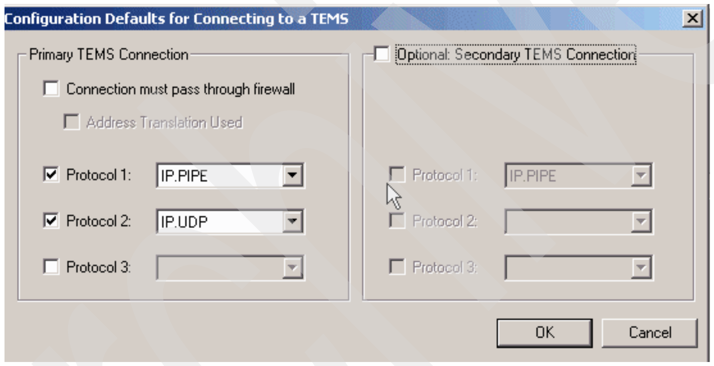
    Specify the default values for any IBM Tivoli Monitoring component to use when they communicate with the monitoring server.

    a. If agents must cross a firewall to access the monitoring server, select **Connection must pass through firewall**.

    b. Identify the type of protocol that the agents use to communicate with the hub monitoring server. There are four options: IP.UDP, IP.PIPE, IP.SPIPE, or SNA as described in Table 7-5 on page 432.

       Three methods for communication can be specified; this enables setup of backup communication methods. If the method identified as Protocol 1 fails, Protocol 2 is used. If using UDP will not be affected by security policies or firewall restrictions, IP.UDP protocol is recommended.

    c. Click **OK**.



Configuration Defaults for Connecting to a TEMS

Primary TEMS Connection

☐ Connection must pass through firewall

☐ Address Translation Used

☑ Protocol 1: IP.PIPE

☑ Protocol 2: IP.UDP

☐ Protocol 3:

☐ Optional: Secondary TEMS Connection

☐ Protocol 1: IP.PIPE

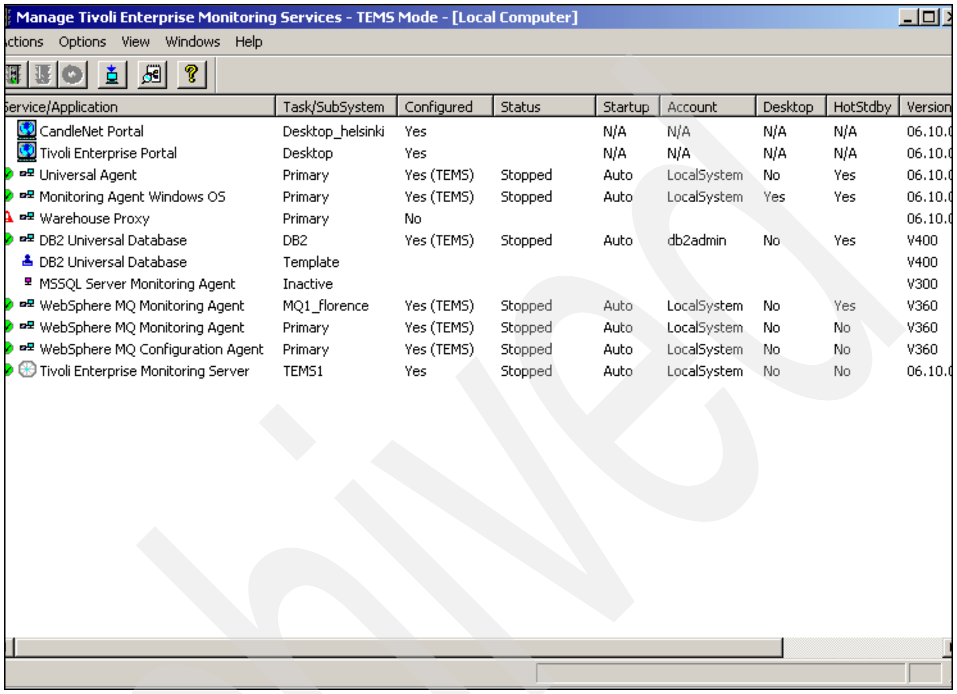
☐ Protocol 2:

☐ Protocol 3:

Figure 7-21 Communication protocol configuration to a TEMS

23. In the next window, click **Finish** to complete the installation.

The Manage Tivoli Enterprise Monitoring Services utility is opened. This utility can be used to start, stop, and configure IBM Tivoli Monitoring components. See Figure 7-22.


Manage Tivoli Enterprise Monitoring Services - TEMS Mode - [Local Computer]
ctions Options View Windows Help

ervice/Application	Task/SubSystem	Configured	Status	Startup	Account	Desktop	HotStdby	Version
CandleNet Portal	Desktop_helsinki	Yes		N/A	N/A	N/A	N/A	06.10.
Tivoli Enterprise Portal	Desktop	Yes		N/A	N/A	N/A	N/A	06.10.
Universal Agent	Primary	Yes (TEMS)	Stopped	Auto	LocalSystem	No	Yes	06.10.
Monitoring Agent Windows OS	Primary	Yes (TEMS)	Stopped	Auto	LocalSystem	Yes	Yes	06.10.
Warehouse Proxy	Primary	No						06.10.
DB2 Universal Database	DB2	Yes (TEMS)	Stopped	Auto	db2admin	No	Yes	V400
DB2 Universal Database	Template							V400
MSSQL Server Monitoring Agent	Inactive							V300
WebSphere MQ Monitoring Agent	MQ1_florence	Yes (TEMS)	Stopped	Auto	LocalSystem	No	Yes	V360
WebSphere MQ Monitoring Agent	Primary	Yes (TEMS)	Stopped	Auto	LocalSystem	No	No	V360
WebSphere MQ Configuration Agent	Primary	Yes (TEMS)	Stopped	Auto	LocalSystem	No	No	V360
Tivoli Enterprise Monitoring Server	TEMS1	Yes	Stopped	Auto	LocalSystem	No	No	06.10.

Figure 7-22 IBM Tivoli Monitoring 6.1 services window

Use this same procedure to upgrade remote CMS systems. When the installation is complete, the startup setting is updated to be automatic and the services are in “started” status.

Upgrading a UNIX CMS

Example 7-1 shows step-by-step how to upgrade a UNIX remote CMS. The sample server is an AIX server named elpaso.

Example 7-1 Upgrading remote CMS to remote TEMS on UNIX server

```
[root@elpaso][/mount]-> ./install.sh

ITM home directory "/candle" already exists.
OK to use it [ y or n; "y" is default ]?  y

Select one of the following:

1) Install products to the local host.
```

```
2) Install products to depot for remote deployment (requires TEMS).
3) View readme files.
4) Exit install.

Please enter a valid number:  1
Initializing ...
Software Licensing Agreement
1. Czech
2. English
3. French
4. German
5. Italian
6. Polish
7. Portuguese
8. Spanish
9. Turkish

Please enter the number that corresponds to the language
you prefer.

2
Software Licensing Agreement
Press Enter to display the license agreement on your
screen. Please read the agreement carefully before
installing the Program. After reading the agreement, you
will be given the opportunity to accept it or decline it.
If you choose to decline the agreement, installation will
not be completed and you will not be able to use the
Program.


International Program License Agreement

Part 1 - General Terms

BY DOWNLOADING, INSTALLING, COPYING, ACCESSING, OR USING
THE PROGRAM YOU AGREE TO THE TERMS OF THIS AGREEMENT. IF
YOU ARE ACCEPTING THESE TERMS ON BEHALF OF ANOTHER PERSON
OR A COMPANY OR OTHER LEGAL ENTITY, YOU REPRESENT AND
WARRANT THAT YOU HAVE FULL AUTHORITY TO BIND THAT PERSON,
COMPANY, OR LEGAL ENTITY TO THESE TERMS. IF YOU DO NOT
AGREE TO THESE TERMS,

- DO NOT DOWNLOAD, INSTALL, COPY, ACCESS, OR USE THE
PROGRAM; AND

- PROMPTLY RETURN THE PROGRAM AND PROOF OF ENTITLEMENT TO

Press Enter to continue viewing the license agreement, or,
Enter "1" to accept the agreement, "2" to decline it or
"99" to go back to the previous screen.

1
Preparing to install the IBM Global Security Kit (GSkit)
+---------------------------------------------------------------------------+
                    Pre-installation Verification...
+---------------------------------------------------------------------------+
Verifying selections...done
Verifying requisites...done
Results...

SUCCESSES
---------
  Filesets listed in this section passed pre-installation verification
  and will be installed.

  Selected Filesets
  -----------------
  gskjt.rte 7.0.3.18                          # AIX Certificate and SSL Java...
  gskta.rte 7.0.3.18                          # AIX Certificate and SSL Base...

  << End of Success Section >>

FILESET STATISTICS
------------------
    2  Selected to be installed, of which:
        2  Passed pre-installation verification
  ----
    2  Total to be installed

+---------------------------------------------------------------------------+
                         Installing Software...
+---------------------------------------------------------------------------+

installp: APPLYING software for:
        gskjt.rte 7.0.3.18


. . . . . << Copyright notice for gskjt >> . . . . . . .
 Licensed Materials - Property of IBM

 999999999
   (C) Copyright International Business Machines Corp. 1996, 2002, 2003.
   (C) Copyright Netscape Communications Corporation.  1995.
   (C) Copyright RSA Laboratories, a division of RSA Data Security, Inc. 1991.

 All rights reserved.
 US Government Users Restricted Rights - Use, duplication or disclosure
 restricted by GSA ADP Schedule Contract with IBM Corp.
. . . . . << End of copyright notice for gskjt >>. . . .

Filesets processed:  1 of 2  (Total time:  4 secs).

installp: APPLYING software for:
        gskta.rte 7.0.3.18


. . . . . << Copyright notice for gskta >> . . . . . . .
 Licensed Materials - Property of IBM

 999999999
   (C) Copyright International Business Machines Corp. 1996, 2002, 2003.
   (C) Copyright Netscape Communications Corporation.  1995.
   (C) Copyright RSA Laboratories, a division of RSA Data Security, Inc. 1991.

 All rights reserved.
 US Government Users Restricted Rights - Use, duplication or disclosure
 restricted by GSA ADP Schedule Contract with IBM Corp.
. . . . . << End of copyright notice for gskta >>. . . .

--- postinstall ---
Linking suitable library according to the machine
Finished processing all filesets.  (Total time:  16 secs).

+-----------------------------------------------------------------------------+
                                Summaries:
+-----------------------------------------------------------------------------+

Installation Summary
--------------------
Name                        Level           Part        Event       Result
-------------------------------------------------------------------------------
gskjt.rte                   7.0.3.18        USR         APPLY       SUCCESS
gskta.rte                   7.0.3.18        USR         APPLY       SUCCESS
 Will enable automatic agent initiation after reboot.
Enter a 32-character encryption key, or just press Enter to use the default
        Default = IBMTivoliMonitoringEncryptionKey
....+....1....+....2....+....3..

GSkit encryption key has been set.
Key File directory: /candle/keyfiles

 Product packages are available in /mount/unix

The following products are currently installed in "/candle:"

  CMS Configurator V360R109 @ AIX R4.3.3
  Candle Management Server V350R256 @ AIX R4.3.3/R5.1 (32 bit)
  UNIX Logs Monitoring Agent V201R118 @ AIX R5.1 (32 bit)
  UNIX OS Monitoring Agent V201R224 @ AIX R5.1 (32 bit)/R5.2 (32 bit)
  Universal Agent V410R109 @ AIX R5.1 (32 bit)



The following prerequisites should be installed now:

  Universal Agent Framework V610R100 @ AIX R5.1 (32 bit)

Do you want to install these prerequisites [ y or n; "y" is default ]?y

 ... installing package "ufaix513"; please wait.

 => installed package "ufaix513."

Product packages are available for the following operating systems and
component support categories:

 1) AIX R5.1 (32 bit)
 2) AIX R5.1 (64 bit)
 3) AIX R5.2 (32 bit)
 4) AIX R5.2 (64 bit)
 5) AIX R5.3 (32 bit)
 6) AIX R5.3 (64 bit)
 7) Solaris R10 (32 bit)
 8) Solaris R10 (64 bit)
 9) Solaris R8 (32 bit)
10) Solaris R8 (64 bit)
11) Solaris R9 (32 bit)
12) Solaris R9 (64 bit)

Type the number for the OS or component support category you want, or type "q"
to quit selection
[ number "3" or "AIX R5.2 (32 bit)" is default ]:

You selected number "3" or "AIX R5.2 (32 bit)"

Is the operating system or component support correct [ y or n; "y" is default
]? y

The following products are available for installation:

 1) Monitoring Agent for UNIX Logs  V06.10.00.00
 2) Monitoring Agent for UNIX OS  V06.10.00.00
 3)Summarization and Pruning agent  V06.10.00.00
 4) Tivoli Enterprise Monitoring Server  V06.10.00.00
 5) Tivoli Enterprise Services User Interface  V06.10.00.00
 6) Universal Agent  V06.10.00.00
 7) all of the above

Type the numbers for the products you want to install, or type "q" to
quit selection.
If you enter more than one number, separate the numbers by a comma or
a space.

Type your selections here:  4
The following products will be installed:

  Tivoli Enterprise Monitoring Server  V06.10.00.00

Are your selections correct [ y or n; "y" is default ]? y

 ... installing "Tivoli Enterprise Monitoring Server  V06.10.00.00 for AIX R5.2
(32 bit)"; please wait.

 => installed "Tivoli Enterprise Monitoring Server  V06.10.00.00 for AIX R5.2
(32 bit)."
 ... Initializing database for Tivoli Enterprise Monitoring Server
V06.10.00.00 for AIX R5.2 (32 bit).
Please enter TEMS name: REMOTE_ELPASO
... creating config file "/candle/config/elpaso_ms_REMOTE_ELPASO.config"
... creating file "/candle/tables/REMOTE_ELPASO/glb_site.txt."
... updating "/candle/config/kbbenv"
... verifying Hot Standby.
 ... Tivoli Enterprise Monitoring Server  V06.10.00.00 for AIX R5.2 (32 bit)
initialized.



Do you want to install additional products or product support packages [ y or
n; "n" is default ]? n

 ... postprocessing; please wait.

 ... finished postprocessing.

 Installation step complete.

 As a reminder, you should install product support on each of your TEM
 servers for any agents you have just installed. This is done via the
 "[ITM home]/bin/itmcmd support" command on your TEM servers.

 You may now configure any locally installed IBM Tivoli Monitoring product via
 the "/candle/bin/itmcmd config" command.
```

When this is complete, the remote TEMS can be reconfigured; however, this procedure is not necessary unless the initial remote CMS configuration has to be changed for some reason, because the settings are kept during the upgrade process.

Follow the example described in Example 7-2.

Example 7-2 Configuring a UNIX remote TEMS (REMOTE_ELPASO)

```
[root@elpaso][/candle/bin]-> itmcmd config -S -t REMOTE_ELPASO
Configuring TEMS...

Hub or Remote [*LOCAL or *REMOTE] (Default is: *REMOTE):
Hub TEMS Host Name (Default is: helsinki):

Network Protocol 1 [ip, sna, ip.pipe, or ip.spipe] (Default is: ip.pipe):

     Now choose the next protocol from one of these:
     - ip
     - sna
     - ip.spipe
     - none
Network Protocol 2 (Default is: ip):

     Now choose the next protocol from one of these:
     - sna
     - ip.spipe
     - none
Network Protocol 3 (Default is: none):
IP Port Number (Default is: 1918):
IP.PIPE Port Number (Default is: 1918):
Enter name of KDC_PARTITION (Default is: null):
Enter path and name of KDC_PARTITIONFILE (Default is:
/candle/tables/REMOTE_ELPASO/partition.txt):

Configuration Auditing? [YES or NO] (Default is: YES):
Hot Standby TEMS Host Name (Default is: none):
Enter Optional Primary Network Name or "none" :(Default is: none):
Security: Validate User ? [YES or NO] (Default is: NO):


TEC Event Integration Facility? (Default is: NO):
Disable Workflow Policy/Tivoli Emitter Agent Event Forwarding? (Default is:
NO):
 ... Writing to database file for ms.

*************************

Editor for SOAP hubs list
```

```
*************************

Hubs
##      CMS_Name
1       ip.pipe:REMOTE_ELPASO[1918]

A)dd, R)emove ##, M)odify Hub ##, U)serAccess ##, C)ancel, S)ave/exit: S
... creating config file "/candle/config/elpaso_ms_REMOTE_ELPASO.config"
... creating file "/candle/tables/REMOTE_ELPASO/glb_site.txt."
... updating "/candle/config/kbbenv"
... verifying Hot Standby.
CMS configuration completed...
```

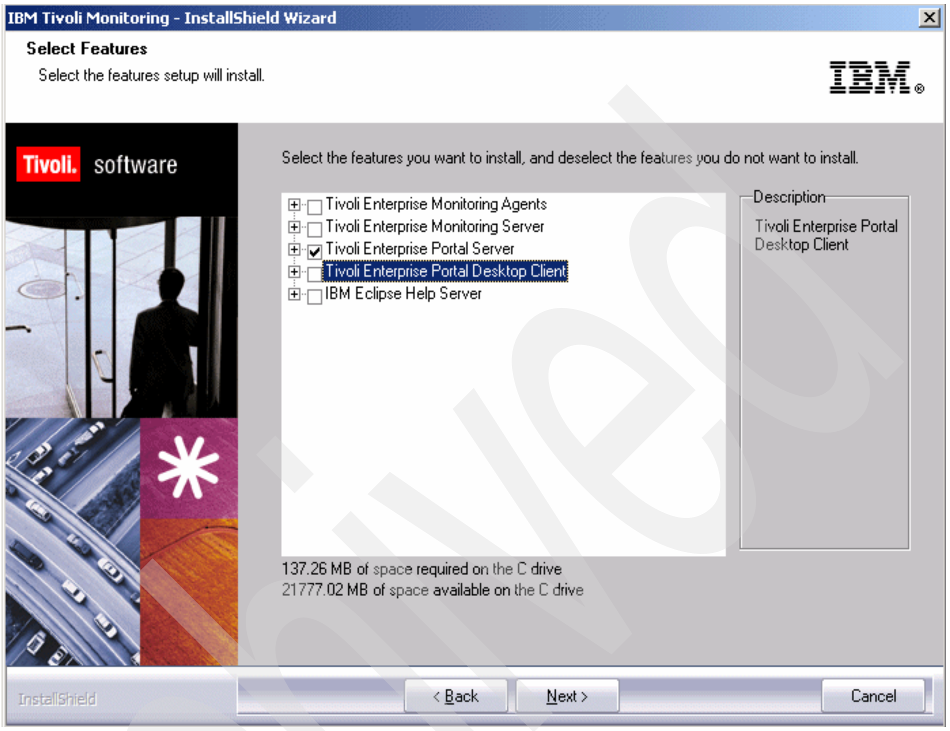
Upgrading the CNPS

Upgrading the CNPS is the next step after having successfully upgraded the Hub and Remote CMS. Remember to make a backup of the CNPS data before proceeding. Refer to “User ID, query, workspace, and navigator configuration backup” on page 418.

Use the following steps to install the Tivoli Enterprise Portal Server on a Windows computer:

1. Launch the installation wizard by double-clicking the **setup.exe** file in the WINDOWS subdirectory of the installation media.
2. Click **Next** on the Welcome window to start the installation.
3. Read and accept the software license agreement by clicking **Accept**.
4. If a suitable database (DB2 or MS SQL) or the IBM Java SDK is not installed on this computer, a message regarding potentially missing required software is displayed. If a proper database configuration is not present, stop the installation, install or upgrade to the required database, and begin the installation again. If the IBM Java SDK is missing, the installation program installs it automatically during a later step.
5. Specify the directory where the portal software and accompanying files will be upgraded. Click **Next**.
6. Type an encryption key to use. This key should be the same as the one used during the installation of the monitoring server to which this portal server will connect (step 5 on page 426). Click **Next** and then **OK** to confirm the encryption key.

7. Select **Tivoli Enterprise Portal Server** from the list of components to install as shown in Figure 7-23.


IBM Tivoli Monitoring - InstallShield Wizard

Select Features

Select the features setup will install.

Select the features you want to install, and deselect the features you do not want to install.

- [ ] Tivoli Enterprise Monitoring Agents
- [ ] Tivoli Enterprise Monitoring Server
- [✓] Tivoli Enterprise Portal Server
- [ ] Tivoli Enterprise Portal Desktop Client
- [ ] IBM Eclipse Help Server

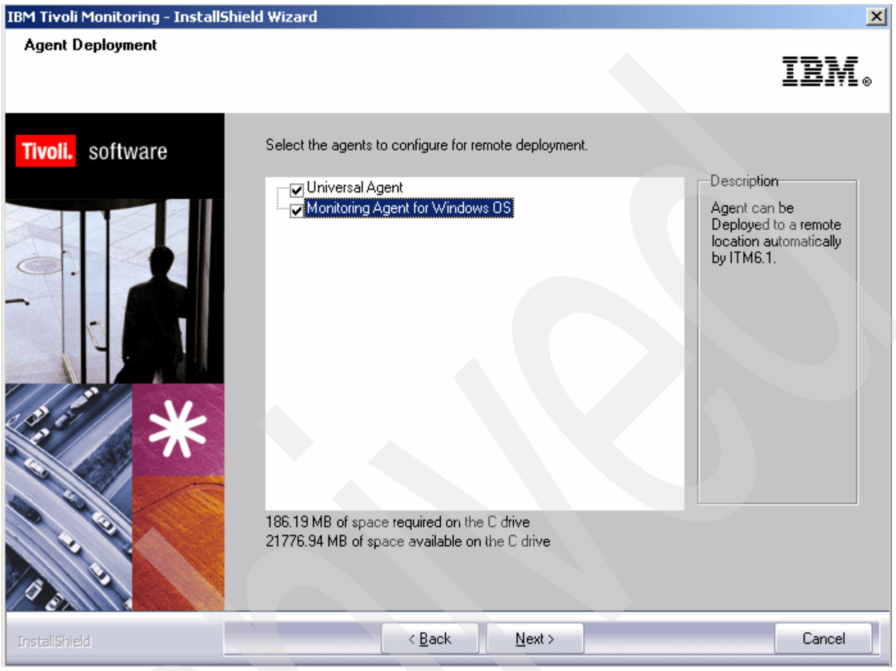
Description: Tivoli Enterprise Portal Desktop Client

137.26 MB of space required on the C drive
21777.02 MB of space available on the C drive

Figure 7-23 IBM Tivoli Monitoring 6.1 Components List

8. If you want to view events from the IBM Tivoli Enterprise Console event server through the Tivoli Enterprise Portal, expand the Tivoli Enterprise Portal Server selection and ensure that **Tivoli Enterprise Console GUI Integration** is selected. Click **Next**.

9. If prompted, select the agents that will be copied to the agent depot for remote deployment later as shown in Figure 7-24; we recommend that you select all of the agents. Click **Next**.



IBM Tivoli Monitoring - InstallShield Wizard

Agent Deployment

Select the agents to configure for remote deployment.

- [✓] Universal Agent
- [✓] Monitoring Agent for Windows OS

Description: Agent can be Deployed to a remote location automatically by ITM6.1.

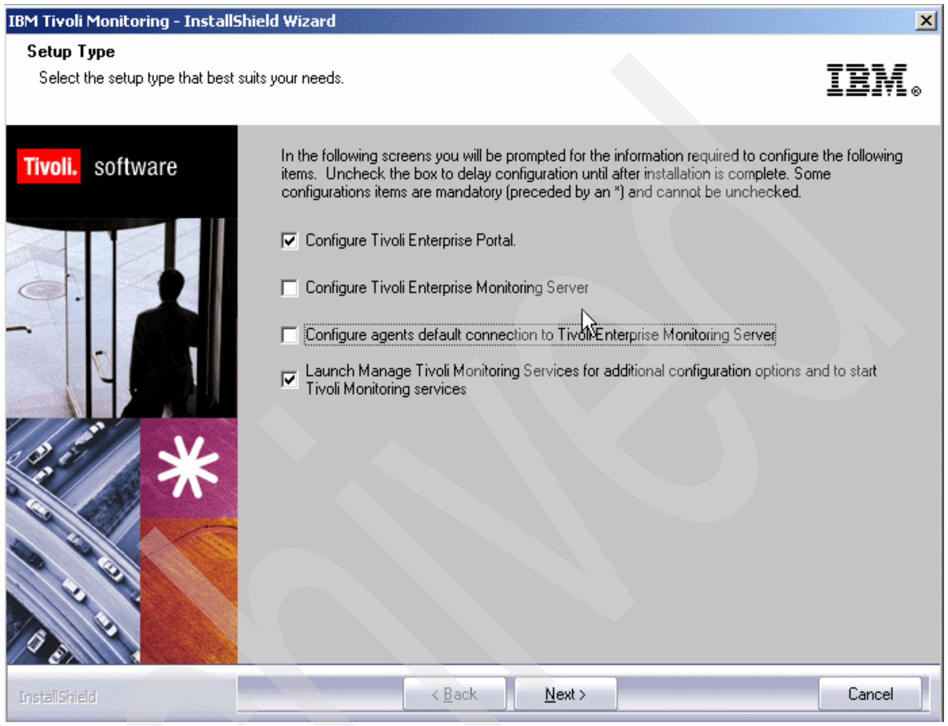
186.19 MB of space required on the C drive
21776.94 MB of space available on the C drive

Figure 7-24 Selected agent for remote deployment

10. Click **Next** on the Review Settings windows.

11. Click **Next** to start the installation.

12. After installation is complete, a configuration window (Figure 7-25) is displayed. Select only the TEPS configuration and the last option to launch the Tivoli Enterprise Monitoring Services console.


IBM Tivoli Monitoring - InstallShield Wizard

Setup Type

Select the setup type that best suits your needs.

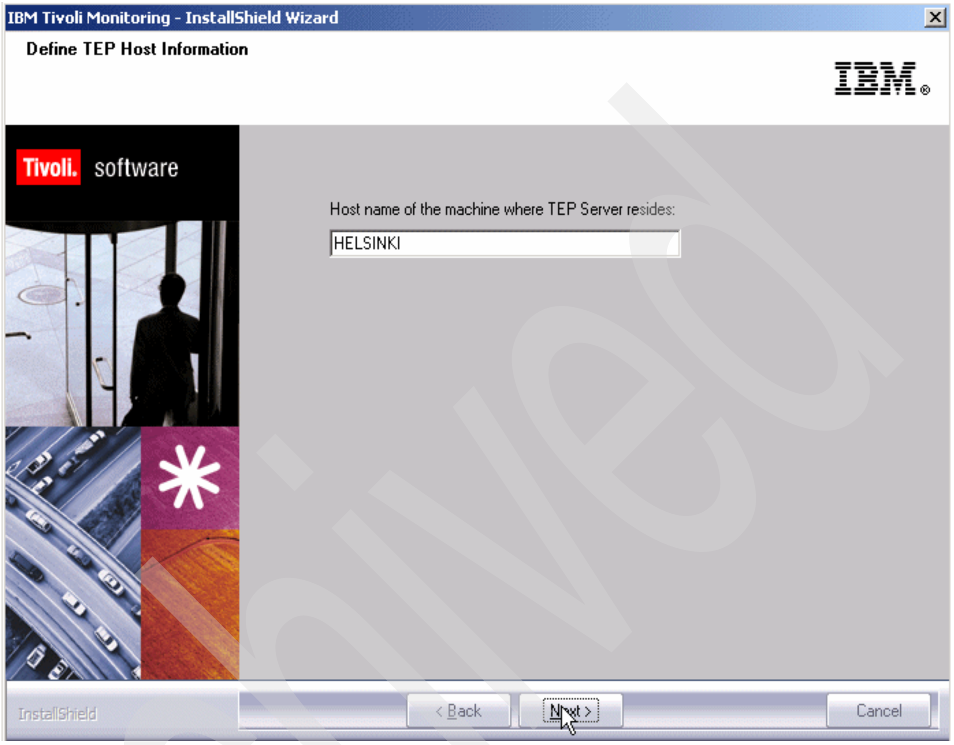
In the following screens you will be prompted for the information required to configure the following items. Uncheck the box to delay configuration until after installation is complete. Some configurations items are mandatory (preceded by an *) and cannot be unchecked.

- [✓] Configure Tivoli Enterprise Portal.
- [ ] Configure Tivoli Enterprise Monitoring Server
- [ ] Configure agents default connection to Tivoli Enterprise Monitoring Server
- [✓] Launch Manage Tivoli Monitoring Services for additional configuration options and to start Tivoli Monitoring services

Figure 7-25 TEPS configuration option window

13.. Click **Next** to begin configuring the portal server and the connection to the monitoring server, and to open Manage Tivoli Monitoring Services.

14. On the next window (Table 7-26) type the host name of the computer where the portal server is being upgraded and click **Next**.


IBM Tivoli Monitoring - InstallShield Wizard

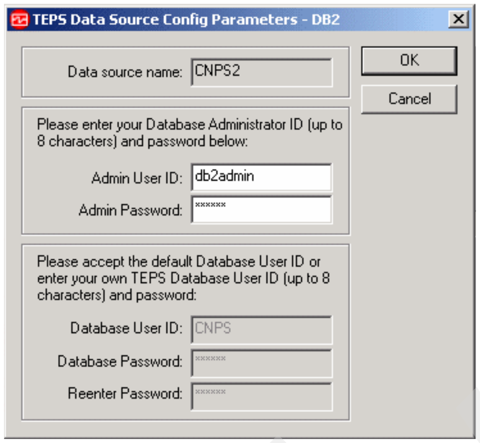
Define TEP Host Information

Host name of the machine where TEP Server resides:

HELSINKI

Figure 7-26 Host name where TEPS will be installed

15. The next window (Figure 7-27) requests the database administrator password. Type it and click **OK**.



TEPS Data Source Config Parameters - DB2

Data source name: CNPS2

Please enter your Database Administrator ID (up to 8 characters) and password below:

Admin User ID: db2admin

Admin Password: ******

Please accept the default Database User ID or enter your own TEPS Database User ID (up to 8 characters) and password:

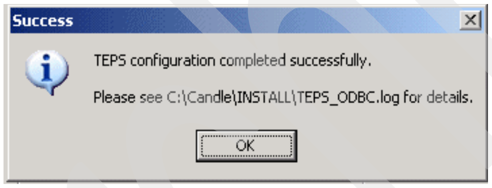
Database User ID: CNPS

Database Password: ******

Reenter Password: ******

Figure 7-27 TEPS database configuration

16. Click **OK** on the message that says that the portal server configuration was successful (Figure 7-28).



Success

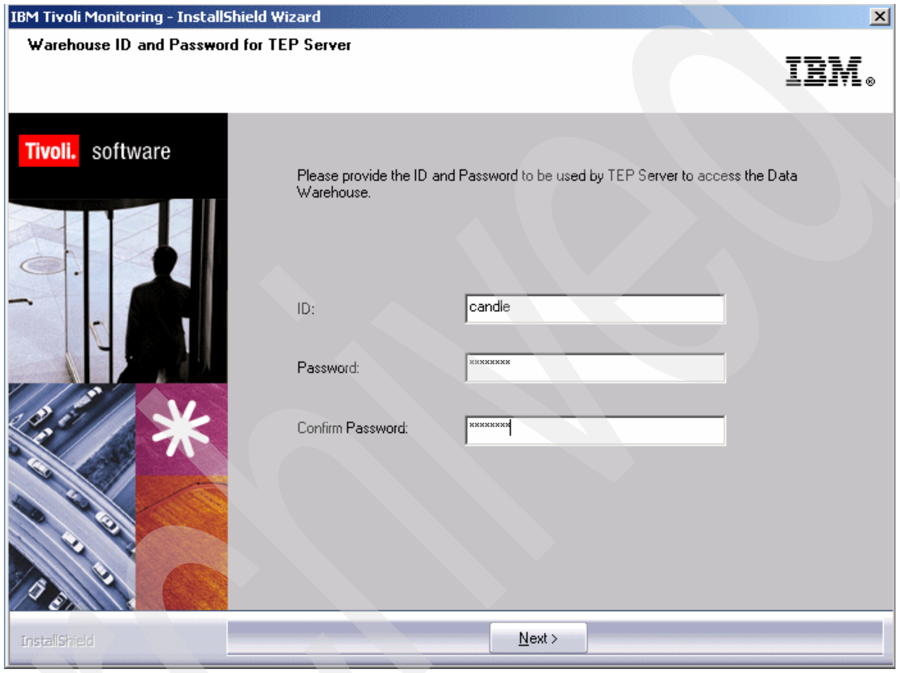
TEPS configuration completed successfully.

Please see C:\Candle\INSTALL\TEPS_ODBC.log for details.

Figure 7-28 TEPS configuration completion window

17. Next it asks about the user credentials to access the Data Warehouse database. The user credentials must match the one used on the Warehouse Proxy database. Type a user ID in the User ID field and type a password for the user in the Password field as shown in Figure 7-29, then click **Next**.

Note: This is the user that owns the Warehouse Proxy database.


IBM Tivoli Monitoring - InstallShield Wizard

Warehouse ID and Password for TEP Server

Please provide the ID and Password to be used by TEP Server to access the Data Warehouse.

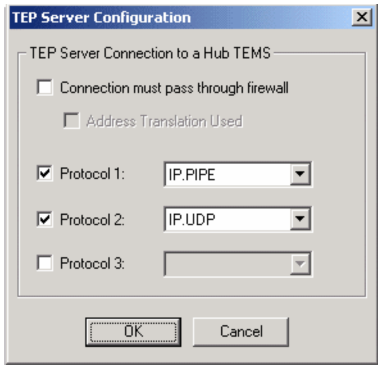
ID: candle

Password: ********

Confirm Password: ********

Figure 7-29 TEPS user configuration

18. Select the communication protocols as shown in Figure 7-30 and click **OK** (Figure 7-30). This defines the values for the connection between the portal server and the hub monitoring server. Refer to Table 3-5 on page 77 for a detailed description of the communication protocols.

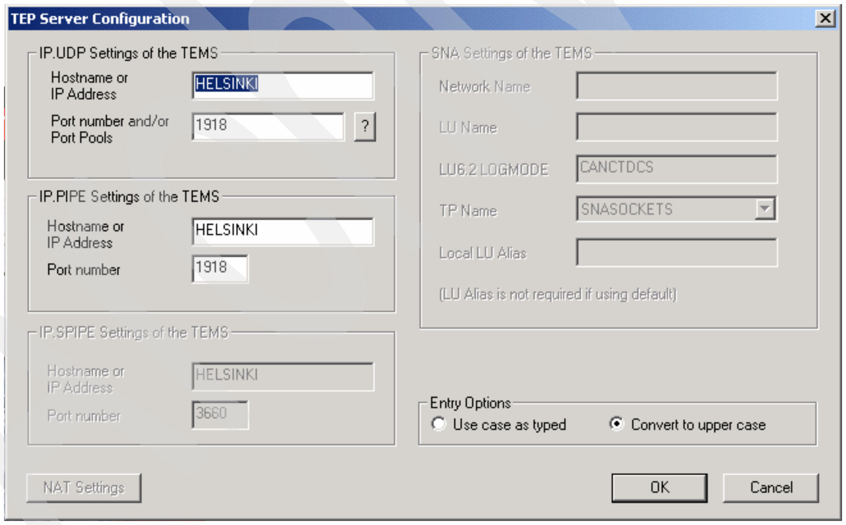

TEP Server Configuration

TEP Server Connection to a Hub TEMS

- [ ] Connection must pass through firewall
- [ ] Address Translation Used
- [✓] Protocol 1: IP.PIPE
- [✓] Protocol 2: IP.UDP
- [ ] Protocol 3:

Figure 7-30 Communication protocol window configuration

19. Type the host name or IP address and the port number for the hub monitoring server as shown in Figure 7-31. Click **OK** when finished.


TEP Server Configuration

IP.UDP Settings of the TEMS
Hostname or IP Address: HELSINKI
Port number and/or Port Pools: 1918

IP.PIPE Settings of the TEMS
Hostname or IP Address: HELSINKI
Port number: 1918

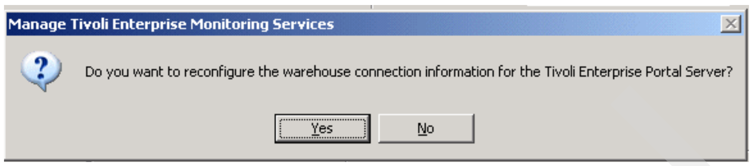
IP.SPIPE Settings of the TEMS
Hostname or IP Address: HELSINKI
Port number: 3660

SNA Settings of the TEMS
Network Name:
LU Name:
LU6.2 LOGMODE: CANCTDCS
TP Name: SNASOCKETS
Local LU Alias:
(LU Alias is not required if using default)

Entry Options
Use case as typed [✓ Convert to upper case]

Figure 7-31 Configuration for connection to the TEMS

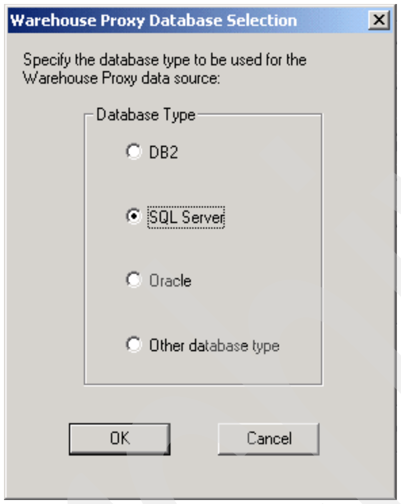
20. Click **OK** on the next window (Figure 7-32) to configure the warehouse connection for the TEP.


Manage Tivoli Enterprise Monitoring Services

Do you want to reconfigure the warehouse connection information for the Tivoli Enterprise Portal Server?

Figure 7-32 Configuring TEPS to connect to the warehouse proxy database

21. Select your warehouse database type and click **OK** as shown in Figure 7-33.


Warehouse Proxy Database Selection

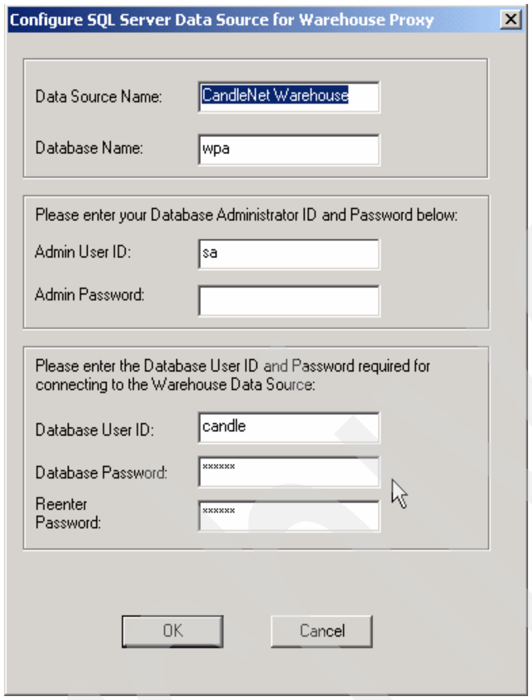
Specify the database type to be used for the Warehouse Proxy data source:

Database Type

○ DB2
● SQL Server
○ Oracle
○ Other database type

Figure 7-33 Selecting the warehouse database type TEPS will connect to

22. Provide the data source name (it must be `CandleNet Warehouse`), the warehouse proxy database name, the Admin user ID (`sa` for MSSQL and `db2admin` for DB2), and the Admin password of the database; and, the database User ID and password for the warehouse proxy database. Refer to the example shown in Figure 7-34.


Configure SQL Server Data Source for Warehouse Proxy

Data Source Name: CandleNet Warehouse

Database Name: wpa

Please enter your Database Administrator ID and Password below:

Admin User ID: sa

Admin Password:

Please enter the Database User ID and Password required for connecting to the Warehouse Data Source:

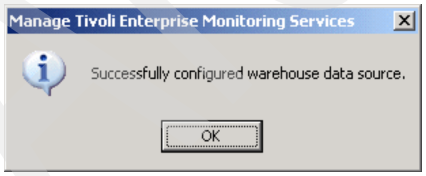
Database User ID: candle

Database Password: ******

Reenter Password: ******

Figure 7-34 Configuring the warehouse proxy database settings for the TEPS

23. Click **OK** on the success window (Figure 7-35).


Manage Tivoli Enterprise Monitoring Services

Successfully configured warehouse data source.

Figure 7-35 TEPS connection to warehouse proxy database successful

24. Click **Finish** to close the installation wizard. After the installation completes, a readme file about Tivoli Enterprise Portal configuration is displayed. Read it and close the window.

25. The Manage Tivoli Enterprise Services console opens with all of the components restarted.

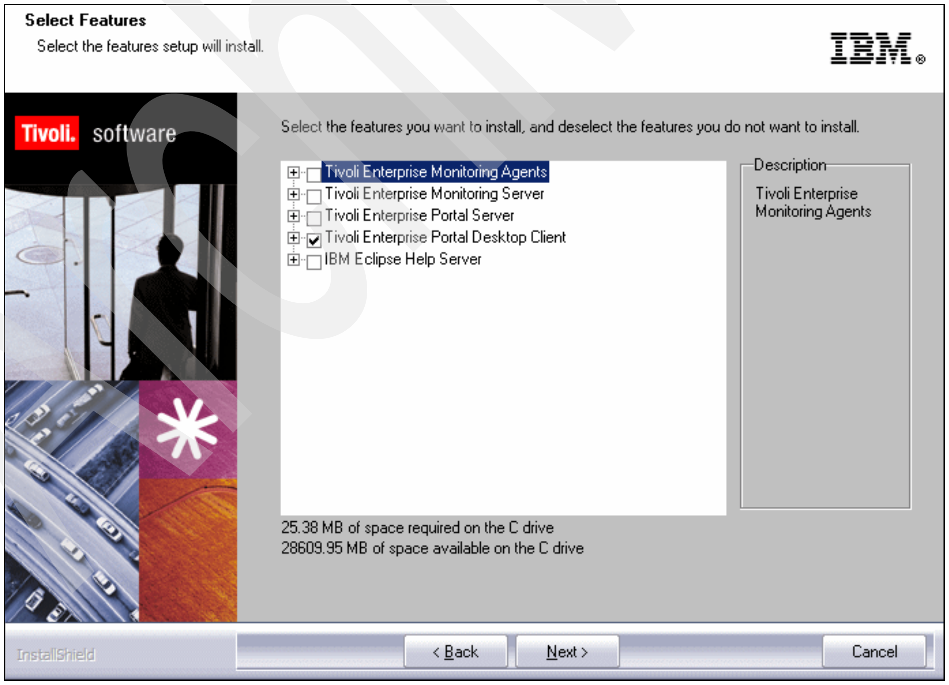
Important: Before attempting to connect to the TEPS, the CandleNet Portal client must be upgraded. Trying to connect to the TEPS using the older CNP version will result in a failed user authentication.

Now the Hub TEMS, Remote TEMS, and the TEPS are installed and configured. The next step is to upgrade the CandleNet Portal.

Upgrading the CandleNet Portal

To function properly, the CNP client must be upgraded to the corresponding IBM Tivoli Monitoring 6.1 Tivoli Enterprise Portal (TEP) client.

1. Launch the installation wizard by double-clicking the **setup.exe** file in the WINDOWS subdirectory of the installation media.
2. Click **Next** on the Welcome window to start the installation.
3. Read and accept the software license agreement by clicking **Accept**.
4. Select the **Tivoli Enterprise Portal Desktop Client** option and choose **Next**.



Select the features you want to install, and deselect the features you do not want to install.

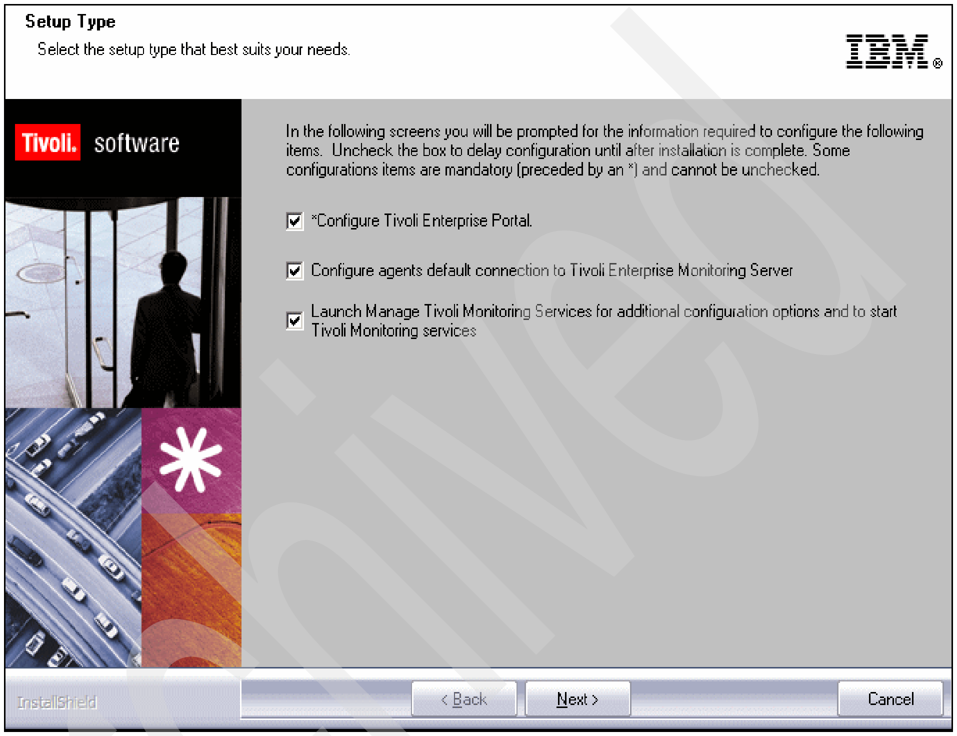
- [ ] Tivoli Enterprise Monitoring Agents
- [ ] Tivoli Enterprise Monitoring Server
- [ ] Tivoli Enterprise Portal Server
- [✓] Tivoli Enterprise Portal Desktop Client
- [ ] IBM Eclipse Help Server

Description: Tivoli Enterprise Monitoring Agents

25.38 MB of space required on the C drive
28609.95 MB of space available on the C drive

Figure 7-36 Select the TEP component to be installed

5. On the next window, type the encryption key or leave the default and click **Next** and **OK**.
6. Click **Next** on the next window (Figure 7-37) to configure the TEP and the connection to the TEMS.



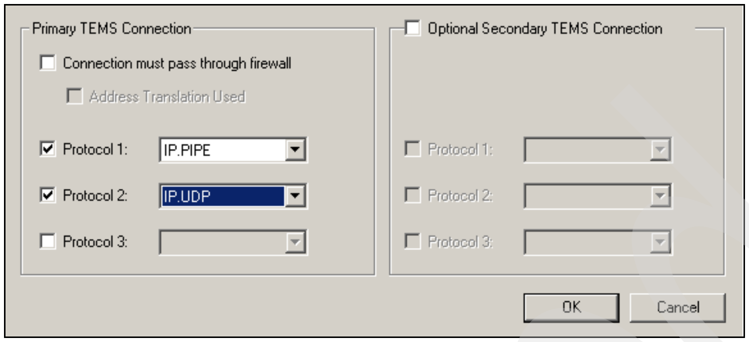
In the following screens you will be prompted for the information required to configure the following items. Uncheck the box to delay configuration until after installation is complete. Some configurations items are mandatory (preceded by an *) and cannot be unchecked.

- [✓] *Configure Tivoli Enterprise Portal
- [✓] Configure agents default connection to Tivoli Enterprise Monitoring Server
- [✓] Launch Manage Tivoli Monitoring Services for additional configuration options and to start Tivoli Monitoring services

Figure 7-37 Choosing the components to be configured

7. Specify the TEPS host name on the next window then click **Next.**

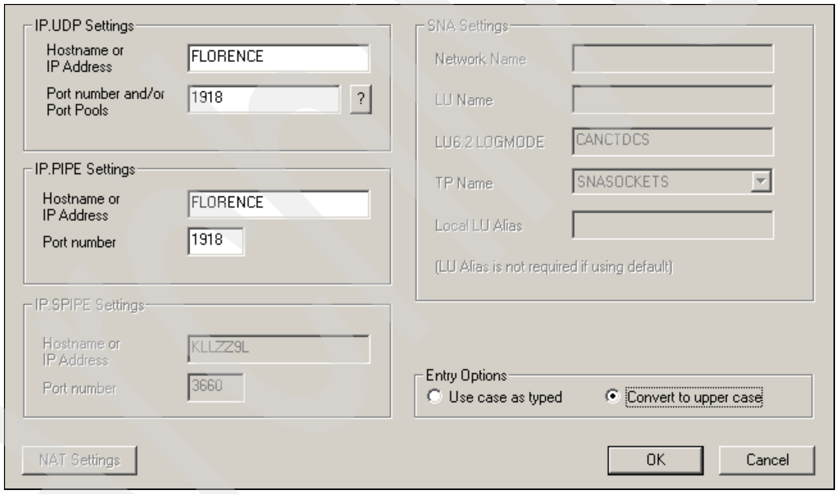
8. Select the communication protocol used between the installed agent with the TEMS as shown in Figure 7-38 and the secondary TEMS if any.


Primary TEMS Connection
Connection must pass through firewall
Address Translation Used
Optional Secondary TEMS Connection
Protocol 3:
OK
Cancel

[✓] Protocol 1: IP.PIPE
[✓] Protocol 2: IP.UDP

Figure 7-38 Setting the communication protocol

9. Type the TEMS host name and port number as shown in Figure 7-39, and click **OK**.



IP.UDP Settings
Hostname or IP Address: FLORENCE
Port number and/or Port Pools: 1918

IP.PIPE Settings
Hostname or IP Address: FLORENCE
Port number: 1918

IP.SPIPE Settings
Hostname or IP Address: KLLZZ9L
Port number: 3660

LU6.2 LOGMODE: CANCTDCS
TP Name: SNASOCKETS

Entry Options: Use case as typed [✓ Convert to upper case]

Figure 7-39 Selecting the TEMS host name and ports

10. Click **Finish**.

This completes the CNP upgrade. The Manage Tivoli Monitoring Services console will open afterward, showing the status.

Upgrading and configuring the agents

Before upgrading agents, certain tasks have to be performed, as described in this section.

Configuration settings for upgraded agents

When OMEGAMON V350 or 360 agents are upgraded to IBM Tivoli Monitoring, agent-specific configuration settings regarding the connection to the monitoring server are not maintained. The IBM Tivoli Monitoring installation uses the default settings for all upgraded agents. To ensure that the upgraded agents can connect immediately to the monitoring server, ensure that the default settings for all agents point to a new monitoring server prior to upgrading. To change the default settings for all agents, right-click an agent in Manage Candle Services and click **Set defaults for all agents**.

Upgrading the Candle OMEGAMON agents

Use the procedure described in 3.2.8, "Tivoli Enterprise Monitoring Agent" on page 95 and 3.2.9, "Deploying TEMA from command line interface" on page 124 to upgrade Candle OMEGAMON agents to IBM Tivoli Monitoring 6.1 agents.

Installing the Summarization and Pruning agent

The Summarization and Pruning agent uses information that is stored in the data warehouse to generate, store, and prune data. The data in the data warehouse is a historical record of activity and conditions in the enterprise. Summarizing the data is the process of aggregating historical data into time-based categories such as hourly, daily, weekly, and so on. It can summarize data to perform historical analysis of the data over time. Pruning data means that old data is deleted automatically, rather than manually. Pruning criteria can be configured to remove old data from the data warehouse to limit the size of the database tables.

During installation of the Summarization and Pruning agent, default values in the KSYENV file populate configuration settings. To change the system-wide default settings for data summarization or pruning configurations, use the Manage Tivoli Enterprise Monitoring Services window to make changes, which affect all agents and attribute groups.

Follow the procedure described in 3.2.13, "Summarization and Pruning agent installation and configuration" on page 139 to install the Summarization and Pruning agent.

Note: Do not configure or restart the Summarization and Pruning agent before upgrading the Warehouse Proxy agent. The Summarization and Pruning agent is installed at this stage only because the warehouse migration tool files are provided by the Summarization and Pruning agent installation.

Upgrading the Warehouse Proxy

Existing data in an OMEGAMON 360 data warehouse can be migrated to the IBM Tivoli Monitoring Tivoli Data Warehouse using a migration tool provided with IBM Tivoli Monitoring 6.1. Before migrating to the new data warehouse, a new ODBC data source must be created for the 6.1.0 version of the Warehouse Proxy agent, and that data source must be configured to connect to a new database. The migration tool is used to move the data from the existing CandleNet Warehouse database to the new IBM Tivoli Monitoring 6.1 database.

The warehouse migration tool files are installed when the Summarization and Pruning agent is installed. Thus, before performing the warehouse proxy upgrade, first install the Summarization and Pruning agent as described in 3.2.13, “Summarization and Pruning agent installation and configuration” on page 139. The following files are installed in the itm_installdir/tmaitm6 directory:

- khdmig.jar
- KHDENV_MIG
- migratewarehouse.bat

It uses the jar files installed under %CANDLE_HOME%/tmaitm6 by the Summarization and Pruning agent:

- cnp.jar
- cnp_vbjorball.jar
- kjrall.jar
- util.jar
- browser.jar

Note: Because the Warehouse Proxy is supported only on Windows, the Warehouse migration tool is supported only on Windows as well.

The following sections provide information about using the warehouse migration tool.

Pre-migration tasks

When migrating from a Candle V350 or V360 environment, a new database and a new ODBC Data Source for that database must be created. The new database has to be separate from the original database because the rules for generating the table names and table column names have changed from the CandleNet Warehouse Proxy to IBM Tivoli Monitoring 6.1. For instance, in V350 the slash character (/) was removed from a column name, whereas in IBM Tivoli Monitoring 6.1, it exists because we are using the Identifier Qualifier to create the table and column name. Also, many table and column names had to be reassigned to be able to support the new database types (Oracle and DB2) due to table name and column name length limits that differ in the databases.

A new table called WAREHOUSEID contains all of the table names or column names that have been converted from a long name to a short name to follow the database vendor specifications. If the new database is Oracle or DB2, it has to be created as a UTF8 database. A tool (migratewarehouse.bat) has been developed to migrate the data from the old data warehouse database to the new data warehouse database.

Perform the following tasks before running the migration tool:

1. Back up the existing Warehouse database.
2. If the existing database is a Microsoft SQL database, install a Microsoft SQL JDBC driver on the computer where that database is installed. The JDBC driver can be downloaded from the Microsoft Web site for Microsoft SQL at:

   `http://www.microsoft.com/sql/downloads/default.asp`
3. Create a new IBM Tivoli Monitoring 6.1 database.
4. Create the ODBC Data Source to connect to that database.
5. Install the IBM Java JRE Version 1.4.2. This can be downloaded externally at:

   `http://www.ibm.com/developerworks/java/jdk`

   The installer from the Windows installation media for IBM Tivoli Monitoring 6.1 could also be run from the WINDOWS\.InIBMJRE subdirectory to install the V1.4.2 JRE.
6. If data has been collected for multiple agents in the past, and migration of the data for all of those agents is desired, then the new Version 6.1 agents must be installed on the Tivoli Enterprise Portal Server.
7. Start the Tivoli Enterprise Portal Server.
8. Stop the Summarization and Pruning agent and the Warehouse Proxy agent.
9. Do not start data summarization with the Summarization and Pruning agent prior to running this migration tool. If this is done, the information that was migrated will not be pruned or summarized. If this is the case, delete the content of the WAREHOUSEMARKER table to enable pruning of both migrated and new data.
10. Remove any table from the source database that does not belong to the Warehouse Proxy agent, because stray tables in the database can lead to migration errors; these can be exported and imported later if needed.

    Important: Also, remove from the source database the WAREHOUSELOG table if it exists. If this step is not performed, it will cause a failed database migration.

Configuring the migration tool

Before running the migration tool, use the following steps to configure the tool:

1. If the JDBC driver was installed on a different directory from the default, update the migratewarehouse.bat file to set the correct CLASSPATH for the JDBC driver. The file contains the following default installation directories for the three supported drivers. Update the location based on where the JDBC driver is installed.
   - Microsoft SQL Server:
     C:\Program Files\Microsoft SQL Server 2000 Driver for JDBC\lib\msbase.jar
     C:\Program Files\Microsoft SQL Server 2000 Driver for JDBC\lib\mssqlserver.jar
     C:\Program Files\Microsoft SQL Server 2000 Driver for JDBC\lib\msutil.jar
   - DB2: C:\Program Files\IBM\SQLLIB\java\db2java.zip
   - Oracle: C:\oracle\ora92\jdbc\lib\classes12.zip
2. Update the variables shown in Table 7-6 in the KHDENV_MIG file.

Table 7-6 KHDENV_MIG file variables for warehouse migration

Type of information	**Variable**	**Description**
Maximum committed rows per transaction	KHD_MAX_ROWS_PER_TRANSACTIONS	The number of rows committed at each insertion of data.
Maximum bad rows skipped before migration is stopped	KHD_MAX_ROWS_SKIPPED_PER_TABLE	The number of rows per table the tool skips to migrate if the data that should be inserted is incorrect. When this number is reached, the tool stops the migration of the table.
Tivoli Enterprise Portal connection information	KHD_CNP_SERVER_HOST	The name of the Tivoli Enterprise Portal Server.
	KHD_CNP_SERVER_PORT	The port number of the Tivoli Enterprise Portal Server.

Type of information	Variable	Description
Source database connection information	KHD_SOURCE_JDBC_DRIVER	The JDBC driver used to connect to the source database.
	KHD_SOURCE_URL	The URL to use to connect to the source database.
	KHD_SOURCE_DATABASE_SCHEMA	The owner or schema in the Candle tables in the source database.
	KHD_SOURCE_DATABASE_USER	The user that connects to the source database.
	KHD_SOURCE_DATABASE_PASSWORD	The password for the user to connect to the source database.
Target database connection information	KHD_TARGET_JDBC_DRIVER	The JDBC driver used to connect to the target database.
	KHD_TARGET_URL	The URL to use to connect to the target database.
	KHD_TARGET_DATABASE_SCHEMA	The owner or schema in the tables in the target database.
	KHD_TARGET_DATABASE_USER	The user that connects to the target database.
	KHD_TARGET_DATABASE_PASSWORD	The password for the user to connect to the target database.

For JDBC driver information, use the syntax appropriate to the driver vendor:

- DB2
  - URL: jdbc:db2:<database>
  - Driver: COM.ibm.db2.jdbc.app.DB2Driver
- MS SQL
  - Server URL: jdbc:microsoft:sqlserver://<server>:<port>;DatabaseName=<database>
  - Driver: com.microsoft.jdbc.sqlserver.SQLServerDriver
  - Default port: 1433
- Oracle
  - URL: jdbc:oracle:thin:@<server>:<port>:<database>

– Driver: oracle.jdbc.driver.OracleDriver

– Default port: 1521

Example 7-3 shows an KHDENV_MIG file configuration set for server “florence,” with the following parameters:

- TEPS host name = helsinki
- TEPS port = 1920
- Source and target database = MSSQL server
- DB Source user = candle350
- DB Source user password = candle
- Source database schema = candle350
- Target database schema = ITMUser
- DB Source user = ITMUser
- DB Source user password = itmpswd1

Example 7-3 KHDENV_MIG file configuration example

```
KHD_MAX_ROWS_PER_TRANSACTION=1000
KHD_MAX_ROWS_SKIPPED_PER_TABLE=300

*  CNP SERVER
*
KHD_CNP_SERVER_HOST=HELSINKI.ITSC.AUSTIN.IBM.COM
KHD_CNP_SERVER_PORT=1920
KHD_SOURCE_JDBC_DRIVER=com.microsoft.jdbc.sqlserver.SQLServerDriver
KHD_SOURCE_URL=jdbc:microsoft:sqlserver://florence:1433;DatabaseName=wpa350
KHD_SOURCE_DATABASE_SCHEMA=candle350
KHD_SOURCE_DATABASE_USER=candle350
KHD_SOURCE_DATABASE_PASSWORD=candle
*
* TARGET WAREHOUSE DATABASE
*
KHD_TARGET_JDBC_DRIVER=com.microsoft.jdbc.sqlserver.SQLServerDriver
KHD_TARGET_URL=jdbc:microsoft:sqlserver://florence:1433;DatabaseName=wpa61
KHD_TARGET_DATABASE_SCHEMA=ITMUser
KHD_TARGET_DATABASE_USER=ITMUser
KHD_TARGET_DATABASE_PASSWORD=itmpswd1
```

After configuring the database setting on the CANDLEHOME\tmaitm6\KHDENV_MIG, edit the CANDLEHOME\tmaitm6\migratewarehouse.bat. Find this line:

```
%_JAVA_CMD% -noverify -classpath "%CLASSPATH%" -Dkjr.trace.mode=LOCAL
-Dkjr.trace.file="%CANDLE_HOME%\TMAITM6\LOGS\KHDRAS1_Mig.log"
-Dcnp.encoding.codeset="UTF8" -Dkjr.trace.params=ALL
-Dibm.itm.khd.productCode=KHD -Dibm.itm.khd.workDir="%CANDLE_HOME%"
-Dibm.itm.khd.mig.common.detailTraceLevel=2 com.ibm.itm.khd.mig.Enable
```

Change it to this:

```
%_JAVA_CMD% -noverify -classpath "%CLASSPATH%" -Dkjr.trace.mode=LOCAL
 -Dkjr.trace.file="%CANDLE_HOME%\TMAITM6\LOGS\KHDRAS1_Mig.log"
 -Dcnp.encoding.codeset="UTF8" -Dkjr.trace.params=ERROR
-Dibm.itm.khd.productCode=KHD -Dibm.itm.khd.workDir="%CANDLE_HOME%"
-Dibm.itm.khd.mig.common.detailTraceLevel=2 com.ibm.itm.khd.mig.Enable
```

Note: Notice that if `-Dkjr.trace.params=ERROR` is set to ALL instead, it can result in extremely large log files. This issue will be addressed in IBM Tivoli Monitoring Version 6.1.1.

Running the warehouse migration tool

After updating the files as described in the previous section, run the migratewarehouse.bat file. After the tools finish running, check the log files KHDRAS1_Mig_Detail.log and KHDRAS1_Mig.log in CANDLEHOME/tmaitm6/logs/.

If everything is fine, output similar to Example 7-4 appears.

Example 7-4 A summary migratewarhouse.bat output execution

```
C:\Candle\tmaitm6>migratewarehouse.bat
Enable:
  Target table NT_Process migrated
Enable:
  Migrating NT_Server to: NT_Server
Enable:
  Target table NT_Server migrated
............
...............
Enable:
  Migrating product = i5/OS
Enable:
  Migrating OS400_Job_Queue to: OS400_Job_Queue
Enable:
run ended
Output thread terminated
```

Status tables

Part of the migration of data to the new warehouse requires shortening the table and column names if the target database is not a Microsoft SQL database. When a name is shortened, an entry is created inside the WAREHOUSEID table in the new database. View this table to check which table names or column names have been modified.

For each package of rows committed, an entry is also made in the new WAREHOUSEMIGLOG table.

If the migration tool fails during the migration, look at the WAREHOUSEMIGLOG table to identify those tables that migrated successfully.

7.2.5 Post-upgrade tasks

This section describes some post-upgrade tasks.

Configuring the Summarization and Pruning agent

To be able to view historical data from the new IBM Tivoli Monitoring Warehouse Proxy database, configure the Summarization and Pruning agent. Follow the steps described in 3.2.13, “Summarization and Pruning agent installation and configuration” on page 139 and 4.2.2, “Historical data types” on page 192.

Configuring the TEPS to point to the new Warehouse Proxy database

The TEPS was set up to point to the old Candle Warehouse database; reconfigure it to use the new IBM Tivoli Monitoring 6.1 Warehouse Proxy database (if it was not already done in “Upgrading the CNPS” on page 445).

Using existing OMEGAMON agents with IBM Tivoli Monitoring

Existing installed OMEGAMON 350 and 360 agents can be supported using the Tivoli Enterprise Monitoring Server; however, the following restrictions apply:

- New features in IBM Tivoli Monitoring 6.1 (such as remote deployment) are not supported on these agents.
- Agents must use one of the following protocols to communicate with the monitoring server:
  - IP.UDP (formerly TCP/IP)
  - IP.PIPE
  - SNA
- The IP.SPIPE protocol is not supported for OMEGAMON agents.
- An OMEGAMON agent cannot be installed on the same computer as a Tivoli Enterprise Portal Server. To monitor something on the same computer as the portal server, install an IBM Tivoli Monitoring agent, if one is available.

To monitor the OMEGAMON agent, use Manage Candle Services to change the monitoring server to which the agent sends data. To enable the Tivoli Data Warehouse to collect data from these agents (through the Warehouse Proxy), copy the product attribute file to the ATTRLIB directory on the Warehouse Proxy agent.

For full interoperability between OMEGAMON agents and IBM Tivoli Monitoring, install the application support files for these agents on the monitoring server, portal server, and portal desktop client.

7.2.6 Known limitations and workarounds

This section describes some known limitations at the time of writing of this book:

The warehouse migration tool can exit with a Java exception

If the source database contains tables other than the OMEGAMON agent tables, this can cause a Java exception to be generated during the migration tool execution, specifically when the WAREHOUSELOG table is present. The CANDLEHOME\tmaitm6\logs\KHDRAS1_MIG.log file will include a message similar to Example 7-5.

Example 7-5 KHDRAS1_Mig_Detail.log error sample

```
   = 105344 java.lang.NullPointerException  at
   com.ibm.itm.khd.mig.hist.HistConfig.run(HistConfig.java(Compiled Code))
      at com.ibm.itm.khd.mig.MigProduct.getParameters(MigProduct.java:727) at
   com.ibm.itm.khd.mig.MigProduct.setup(MigProduct.java:623) at
   com.ibm.itm.khd.mig.Enable.run(Enable.java:122) at
   com.ibm.itm.khd.mig.Enable.main(Enable.java:69)
   == 105345 EXCEPTION: Failed to execute setup.
== 105346 EXCEPTION: Failed to run
```

Use the following procedure to resolve this issue:

1. Drop the table WAREHOUSELOG from the source database.
2. Drop any tables already migrated in the target database to avoid duplicate data when rerunning the migration tool.
3. Rerun the migration tool.

Failure to migrate rows for OS400 Agent tables

The source tables listed in Table 7-7 have missing columns.

Table 7-7 Source tables having missing columns

Table	Column
OS400_Job	Synch_IO
OS400_Job	End_Status

Table	Column
OS400_Network	Alert_Default_Focal_Point
OS400_IO_Processor	IOP_Bus_Number
OS400_IO_Processor	IOP_Bus_Address

This will cause a migration failure when migrating those three tables. No workaround exists for this specific issue. A defect has been opened to be resolved with IBM Tivoli Monitoring 6.1.1.

How to migrate specific tables that the migration tool failed to migrate to the new warehouse proxy database

During the Warehouse Proxy data migration, some table might have failed to be exported; in that case use the following procedure to export those tables separately.

1. Create a new database.
2. Export only the failed tables from the old original source CandleNet Warehouse database, and import them into the new database that was just created above.
3. Reconfigure the KHDENV_MIG file to use this new database as the source. That way, only the source data that failed to be migrated will be analyzed. Example 7-6 shows how to reconfigure KHDENV_MIG file to point a new database called new_db build in a MSSQL RDBMS on a server called florence; the database is owned by new_user.

Example 7-6 Configuring the KHDENV_MIG to point to a new database

```
KHD_SOURCE_URL=jdbc:microsoft:sqlserver://florence:1433;DatabaseName=new_db
KHD_SOURCE_DATABASE_SCHEMA=new_user
KHD_SOURCE_DATABASE_USER=new_user
KHD_SOURCE_DATABASE_PASSWORD=new_password
```

4. Rerun the migration tool.

8

Real-life scenarios

This chapter provides a blueprint for turning a monitoring need into an enterprise event. We also discuss the GUI console (application or browser) with an in-depth look at the components of the GUI including the Navigator, queries, filters, take action, situation actions, and situations.

In this chapter, the following topics are discussed:

- Who owns the tool?
- Defining the need for monitoring
- Building the monitoring
- Building another example: monitoring Tivoli
- Building a graphical view for a Tivoli administrator

8.1 Who owns the tool?

Before we discuss real-life application of the tool, we should address a basic question. Who will own the tool?

An organization performing Enterprise Systems Management can be identified as somewhere along an evolutionary process for distributed computing. The host arena has maturity. Distributed environments, however, generally can be expected to be less mature. This is not to say that the less mature environment is somehow qualitatively less desirable than a more mature environment, it is merely the recognition of a current state. The location on the maturity scale is a recognition of process maturity, technical achievement, and financial investment.

At a lower level of maturity, we can expect that the Tivoli team will own the monitoring solution and development of monitors for the enterprise. An increasingly common movement among more mature enterprise customers is to turn their developed Tivoli products into services. The Tivoli team manages infrastructure and sets direction, trains for the use of the services, and provides support for those trying to use the services. In this context, for IBM Tivoli Monitoring 6.1, the creation of monitors is left to the administrators and application teams while overall tool function and architecture is held by the Tivoli team.

What are the criteria for determining whether your implementation should be managed this way?

- IBM Tivoli Monitoring 6.1 is already installed in the environment and is stable (including implementation processes). The Tivoli team has acquired IBM Tivoli Monitoring 6.1 skills.
- Administrator teams and application teams have skills required for programming (general skills).
- You are prepared to educate the administration and application teams about how to build their monitors and provide documentation to support this.
- Precedence exists for application teams and administration teams to “own” their own monitors, and they are ready to “own” their monitors (recognizing the resource requirements within their own teams).
- Change control to the production environment is a fairly mature process, and the application teams and administration teams will follow the process.
- Development and QA environments will be available for the application and administration teams to develop their monitors.
- A commitment to a QA process will be followed prior to introduction of the monitors into the production environment.

If these criteria cannot be met in your environment, your organization will be much better served by holding control and development and management of the tool in a single group.

8.2 Defining the need for monitoring

This section discusses the methodology to be used in turning a need for monitoring into a technical solution to answer that need. Non-technical points (process-related) will be made in addition to the stated technical details.

Five distinct types of activities are advocated in this methodology for defining the need for monitoring:

- 8.2.4, “Identification of monitoring need” on page 476 discusses how the nomination should come from either a business problem or from the Problem Management activity within your organization.
- 8.2.5, “Identify the target audience” on page 478 discusses the importance of ownership of the events. The development of monitoring should always be in coordination with the system/application administrators and the help desk.
- 8.2.6, “Identify and refine the possible events list” on page 479 discusses how to come up with a complete solution by investigation of the event request.
- 8.2.7, “Meet with the target audience for approval” on page 480 is a very important activity. At this point, all parties should agree to the escalation process for this event and its owner if changes are required.
- 8.2.8, “Create, test, and implement the monitor” on page 480 discusses the process that should surround these activities.

Without undertaking these activities, the overall success of any event creation will be hampered in some way by acceptance, value realization, or satisfaction of need.

Before we begin to discuss this need for monitoring, we must digress a bit to explain some concepts related to the product that must be understood. These concepts are important to understanding the behavior of the product as it is observable at the console and therefore the usage of the tool in your environment (enterprise impact).

8.2.1 Understanding the terms

For this discussion, you need to understand the following terms as defined in *IBM Tivoli Monitoring, Version 6.1.0, User's Guide, SC*32-9409.

Event	An action or some occurrence, such as running out of memory or completing a transaction, that can be detected by a situation. The event causes the situation to become true and an alert to be issued.
Event indicator	The colored icon that displays over a Navigator item when an event opens for a situation.
Monitor interval	A specified time, scalable to seconds, minutes, hours, or days, for how often the monitoring server checks whether a situation has become true. The minimum monitor interval is 30 seconds; the default is 15 minutes.
Pure event	A pure event occurs automatically, such as a paper-out condition on the printer or writing of a new log entry. Situations written to notify of pure events remain true until they are manually closed or automatically closed by an UNTIL clause.
Sample	The data that the monitoring agent collects for the server instance. The interval is the time between data samplings.
Sampled event	Sampled events happen when a situation becomes true. Situations sample data at regular intervals. When the situation is true, it opens an event, which gets closed automatically when the situation goes back to false (or you can close it manually).
Situation	A set of conditions that are measured according to criteria and evaluated to be true or false. A condition consists of an attribute, an operator such as greater-than or equal-to, and a value. It can be read as `If - `*`system condition`*` - `*`compared to`*` - `*`value`*` - is true`. An example of a situation is `IF - CPU usage - > - 90% - TRUE`. The expression `CPU usage > 90%` is the situation condition.
State	The severity of the situation event: critical, warning, or informational. Indicated by a colored event indicator, state is set in the Situation Editor and can be different for different Navigator items.
Status	The true or false condition of a situation.
View	A windowpane, or frame, in a workspace. It may contain data from an agent in a chart or table, or it may contain a terminal session or browser, for example. A view can be split into two separate, autonomous views.

8.2.2 Pure versus sampled events with impact to the IBM Tivoli Monitoring 6.1 console user

The concept of a pure event can be understood as a stateless event. There is no recognition of a state of a resource in a pure event. An easily understood example is when an event is read from a Windows log file.

In the case of the log file monitor, the agent is merely matching lines from the log that it has been configured to forward to the TEMS. The agent is not keeping track of anything to compare events and there is no evaluation other than matching.

There is also no concept of an interval when building a situation to detect pure events although there is some configuration of a time interval possible for most agents that detect pure events in most cases ... but that is for all operation of the agent and not on a situation-by-situation basis as with sampled events.

The sampled event, in contrast, has a state. The current state of the resource at sample time has a value and a state against which it is being measured. If instead of reading the log for an event we evaluated the current status of the storage application process for up/down, this would be a sampled event.

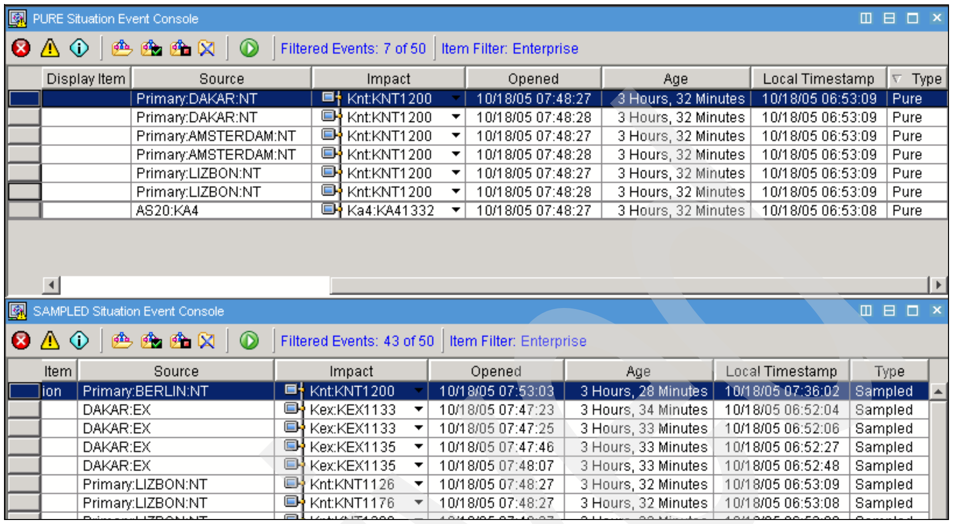
We evaluate the status (what is it) and compare against some criteria (up/down). When the monitor determines that the criteria have been met, the sampled situation becomes true and thus appears on the event console. When it is resolved (or no longer true), it is false.

When a pure event comes to the console, it is there until acted on by a human operator (if you are managing the events from the console) unless you include an UNTIL setting to expire them at a later time. UNTIL is discussed further in 8.3.7, “Using the Until tab” on page 495.

In contrast, when a situation reports a sampled event and it comes to the console, it is there until the conditions of the situation change that would result in it going back to false. Sampled events cannot and should not be closed by the console, by IBM Tivoli Enterprise Console, or by the UNTIL tab in the situation editor.

A small business support team (or any organization without an enterprise view tool) using the events console would need a view created in a workspace for the console operators that displays only pure events for events they are required to action. A second view might display sampled events to alert the console operator to the fact that they cannot close the events but might want to investigate situations that are currently visible in that view. Figure 8-1 on page 474 illustrates such a console.

This is done via the creation of a custom workspace for the console operator. Chapter 13, “Administration and maintenance” on page 697 covers this process.



PURE Situation Event Console

Filtered Events: 7 of 50 | Item Filter: Enterprise

Display Item	Source	Impact	Opened	Age	Local Timestamp	Type
	Primary:DAKAR:NT	Knt:KNT1200	10/18/05 07:48:27	3 Hours, 32 Minutes	10/18/05 06:53:09	Pure
	Primary:DAKAR:NT	Knt:KNT1200	10/18/05 07:48:28	3 Hours, 32 Minutes	10/18/05 06:53:09	Pure
	Primary:AMSTERDAM:NT	Knt:KNT1200	10/18/05 07:48:27	3 Hours, 32 Minutes	10/18/05 06:53:09	Pure
	Primary:AMSTERDAM:NT	Knt:KNT1200	10/18/05 07:48:28	3 Hours, 32 Minutes	10/18/05 06:53:09	Pure
	Primary:LIZBON:NT	Knt:KNT1200	10/18/05 07:48:27	3 Hours, 32 Minutes	10/18/05 06:53:09	Pure
	Primary:LIZBON:NT	Knt:KNT1200	10/18/05 07:48:28	3 Hours, 32 Minutes	10/18/05 06:53:09	Pure
	AS20:KA4	Ka4:KA41332	10/18/05 07:48:27	3 Hours, 32 Minutes	10/18/05 06:53:08	Pure

SAMPLED Situation Event Console

Filtered Events: 43 of 50 | Item Filter: Enterprise

Item	Source	Impact	Opened	Age	Local Timestamp	Type
ion	Primary:BERLIN:NT	Knt:KNT1200	10/18/05 07:53:03	3 Hours, 28 Minutes	10/18/05 07:36:02	Sampled
	DAKAR:EX	Kex:KEX1133	10/18/05 07:47:23	3 Hours, 34 Minutes	10/18/05 06:52:04	Sampled
	DAKAR:EX	Kex:KEX1133	10/18/05 07:47:25	3 Hours, 33 Minutes	10/18/05 06:52:06	Sampled
	DAKAR:EX	Kex:KEX1135	10/18/05 07:47:46	3 Hours, 33 Minutes	10/18/05 06:52:27	Sampled
	DAKAR:EX	Kex:KEX1135	10/18/05 07:48:07	3 Hours, 33 Minutes	10/18/05 06:52:48	Sampled
	Primary:LIZBON:NT	Knt:KNT1126	10/18/05 07:48:27	3 Hours, 32 Minutes	10/18/05 06:53:09	Sampled
	Primary:LIZBON:NT	Knt:KNT1176	10/18/05 07:48:27	3 Hours, 32 Minutes	10/18/05 06:53:08	Sampled

Figure 8-1 A custom workspace with separate pure and sampled event views

8.2.3 Pure versus sampled events and customer impact

If you are an enterprise customer with a central event console such as IBM Tivoli Enterprise Console, pure events are presented and treated in a way that is most closely related to the events that you had received from the Tivoli Enterprise Console Logfile Adapter, the Windows event log adapter, the SNMP adapter, and other related adapters. There is no schedule for the events. They arrive according to the logic of the agent that is sending them. The sampling, in that sense, is done at the agent’s discretion and may vary from agent to agent.

What to do with these events is easily recognized and fits within our traditional notion of enterprise monitoring. Integration with IBM Tivoli Enterprise Console can result in pure events creating IBM Tivoli Enterprise Console events. For more information on the integration with IBM Tivoli Enterprise Console, review 10.2.1, “Installing the OTEA” on page 585.

Important: It is permissible for pure events to be closed by rules from IBM Tivoli Enterprise Console. It is not advisable (although currently there is nothing in our evaluated version of the product that disables it from doing so) to close sampled events. As will be mentioned later, this causes problems as the event is closed while the situation remains true and it does not recover until the situation becomes false and then true again.

While one strategy might be to not generate an IBM Tivoli Enterprise Console event for a sampled situation (not forwarding the events to IBM Tivoli Enterprise Console), this hardly is a maintainable strategy in an enterprise environment, especially because sampled events are probably as common as pure events (or more so).

For sampled events, it is less clear for an enterprise customer (with IBM Tivoli Enterprise Console or another enterprise console product), especially if the intent is to try to use the IBM Tivoli Monitoring 6.1 event console for some adjunct purposes to IBM Tivoli Enterprise Console. The questions start to rise: What is our policy for exploring these sampled situations? When do we decide that a sampled event becomes an IBM Tivoli Enterprise Console event or trouble ticket?

Note: If the decision is made to not use the IBM Tivoli Monitoring 6.1 event console as some adjunct tool to an existing Tivoli Enterprise Console implementation (that is, not installing the IBM Tivoli Enterprise Console event synchronization), the point is moot and all sampled situations should send IBM Tivoli Enterprise Console events and then be set to expire in the UNTIL tab (for pure). You *do* have to install the BAROC file for IBM Tivoli Monitoring 6.1 at the IBM Tivoli Enterprise Console server. You must set up the TEMS to use the IBM Tivoli Enterprise Console event integration facility so that the events are sent to IBM Tivoli Enterprise Console. Refer to Chapter 10, "Integration with IBM Tivoli Enterprise Console" on page 583 for details.

If no one looks at the IBM Tivoli Monitoring 6.1 console, the current status of the situation is not an issue. This puts you at process parity with IBM Tivoli Monitoring 5.x and Distributed Monitoring 3.7. At that point, two-way communication between IBM Tivoli Enterprise Console and IBM Tivoli Monitoring 6.1 is not required or desired. The one-way communication delivers the events.

For customers who choose to use the IBM Tivoli Monitoring 6.1 console in an enterprise environment, we can offer the best practices from some OMEGAMON XE customers. Many who used the OMEGAMON XE product chose to implement some logic for sampled events where possible. This logic is a suggestion about how you could choose to deal with these sampled events.

Your organization should explore the concept of these events with the administrators of the application and systems involved in the sampled events to decide whether this is the appropriate course of action. It is possible that by the time the sampled event indicates an issue and the human being arrives on the scene, the sampled event could have become false again. They need to understand how the product functions in order to reconcile the current state with the fact that they were paged or got a trouble ticket or whatever is the action. We discuss using the console as an evaluative tool later in this chapter.

These steps outline the suggested best practice for dealing with sampled events:

1. When the sampled situation becomes true, attempt to resolve the issue via automation. You will not want operators to try to resolve these issues (as indicated before) until you are sure that the sampled situation will not become false within your tolerance limits.
2. Via policy, wait for the next sample to determine whether the automation has resolved the situation. Even if the event that the situation addresses has been resolved, it will not make the situation false until it samples again. To make this most efficient, the time between samples should allow for the automation to have a chance to resolve the issue. In other words, if the automation will require stopping and restarting a process and it takes x seconds for the process to recycle, the sampling interval should not be less than x seconds.
3. If the situation has not become false after the next iteration, you will want to take action (such as generate a trouble ticket or open an IBM Tivoli Enterprise Console event).

If automation is not possible, you want the policy to notify someone immediately so that a resolution process can be started as soon as possible (according to agreed response times negotiated in your Service Level Agreements.)

8.2.4 Identification of monitoring need

This identification of a need can include things such as recognizing the need for application monitoring for new applications, improving existing monitoring solutions through the introduction of newer, more sophisticated monitoring, or correlation among existing known monitoring items. This identification should identify the criticality of the request and the impact of the situation that this monitoring is to address in order to prioritize development activities across the available resources.

One fairly straightforward way to discover the need for monitoring is through the ITIL concept of a Problem Management function within your organization.

When a situation occurs that is of enough significance in your organization and that could be detected via some technical means, the Problem Management function should request via the application/system owner that some mechanism be created to proactively identify and resolve the issue, identify it and alert the required parties, or take some automated action to attempt to resolve the issue before it becomes an incident.

We do advocate the complete ITIL Service Level Management process set including Help Desk, Incident Management, Change Management, and Release Management: processes that should be observed in the monitoring space and should be recognized as a part of this activity.

The second way that the need for monitoring is discovered should be through the requirement for application management by the organization. All new developed and purchased applications in the environment should be monitored as a part of systems management to the level deemed necessary for the importance of the system or application being monitored.

In the case of off-the-shelf software, the vendor should be able to provide error conditions for trapping and advice as to which conditions are most critical. For developed software, the developer should build the software with monitoring in mind. This means that alerting to possible problems prior to failure should be a function of the application.

Monitoring that is merely reactive (identifying post-failure) can never be proactive in assisting the administrators of the application in preventing failure. Process-down monitoring is important, but far more valuable is a set of alerts to recognize that conditions exist that may precipitate or precede a failure.

Note: As an example of this idea, would you rather have a tornado warning given to you or just wait for it to hit your house to react? As sure as you are of your answer to this question, it is very common for administrators to say to you "don't bother me until it actually goes down." While this does assure them that there is something they can do, it comes at the expense of the service which at that point is compromised.

In general, you will want to go through a planning session to design your monitoring and dashboard views. The following approach is advocated.

Define the objectives of the solution. Make sure to state the business objectives. Define the scope of the solution. Identify the most critical applications.

Examine the existing technology infrastructure to see what needs can be met with the existing solution. If parts will not be met by the existing solution, describe the additional monitoring pieces that have to be created. Identify the information and functions you need to design for the user communities.

Additionally, a gatekeeping function should exist for the team who develops or implements the monitoring solution. This gatekeeper might be the manager of the team or a business analyst role within the team.

The five advocated areas are the criteria that should be used by the gatekeeper to make sure that the resources that are required for building, maintaining, and operating the monitoring solution are appropriately managed by this request (and the other requests that have been or will be made.)

The criticality of the request helps drive the schedule for design and implementation of the monitoring. This is also determined by the impact of the outage or failure that this monitoring addresses and the likelihood of failure. Mission-critical systems and applications should be monitored more than and differently from systems whose impact is low. If an event happens every day and it is related to a mission-critical system, this event should be prioritized much higher for monitoring development than something that has never been known to have occurred, even though it is related to a significant event.

8.2.5 Identify the target audience

Every event that will be monitored in the enterprise should have some known value and an agreed-on course of action. It should be agreed to by all parties from the operations person who views the event to the administrator who receives the problem management ticket and is expected to action the event. The target audience for this event must fully accept and own the implications of this event; otherwise monitoring for this event is a waste of time to even build.

Even so-called “best practices” events that may be implemented by a solution-provider or the vendor of the monitoring solution are not best practices in your organization if they fail to achieve recognition as such. The administrator at company A may fully believe that Event X (an out-of-the-box, solution-implemented event) is very important and action it every time it occurs. The administrator at company B, who does not, promptly disabled the monitor after it alerted him twice in one day. Both experiences and activities may be valid depending on the needs of the organization and their systems administrator’s experience and perceptions.

Tip: We know from experience that ownership is critical. Enabling monitors because they sound “cool” or because they look important (the graph is particularly entrancing) usually results in wasted resources or dissatisfied customers of the monitoring team. It is not easy to force the importance of an event if it has no owner willing to acknowledge its importance and represent it to the organization.

As an example, a monitor is created that monitors the level of activity of the computer system by examining the queue lengths, processor activity levels, and so forth. The event is generated and results in a problem management ticket. Per our advocated item, first, the administrator of that system must agree that activity level is something they want to be alerted about and indeed it is a problem that they action.

If the administrator who receives this event is not willing to action the event (knows the value in being notified and agrees to action it), the reception of the event is a nuisance. In environments where the events were not agreed to be important and understood by all parties there is no clear value in the monitoring solution that can be realized or recognized by the organization.

Such an organization usually has many monitoring solutions, each owned by a different faction of the organization, and each believed to deliver the value that the faction seeks, regardless of the ability of a single tool to meet the needs of the entire organization.

8.2.6 Identify and refine the possible events list

The first step of design is to map the monitoring, managing, and integration facilities to the discovered needs. When you have this mapping design, move into organizing the presentation of the information. What business views, links, workspaces, and reports are required and for which communities?

Plan the event management and automation to solve monitoring needs, always keeping in mind the changes in volume of events and the impact on the support and operations areas. Experts from each area should be pulled in to help with this design.

For every event that is generated by every application and every system in the environment there likely exists some relationship with other events or occurrences in the environment. Consider a request to monitor this event:

```
EVENT_ID 20503        The application WYSIWYG has failed with error -9.
```

This can be seen as a possible mistake to alert when it is found to be a symptom of the WYSIWYG application on server Y failing and server X being unable to contact it. You cannot rely on the vendor to tell you these relationships and often they will be found only with experience. Even though picking up these alerts for the purpose of validation can be a worthwhile endeavor, it should be understood what the event means so that an appropriate correlation can be made and the action taken to resolve the correct issue.

As important as identifying the specific indicators is to determine the approximate event volume in the environment. It is important to make sure that your activity does not overwhelm the monitoring solution that exists (an excessive number of events or time interval) or require significant resources beyond a sustainable or desirable level.

The monitoring team usually works with the team supporting the enterprise tools to assure that their desired monitoring solution is technically feasible and supportable. As an example, an application developer who feels that event 20503 must be trapped to correlate might need some help to understand the impact of doing so if they throw these events at a rate of 100/second. In this case, doing so requires some changes to the architecture and possible other components.

8.2.7 Meet with the target audience for approval

Prior to implementation of the monitor in the enterprise environment, all activities listed above should be completed. This involves such sign-offs as:

- A defined business value has been stated.
- An owner has been identified and they agree to the ownership.
- An appropriate SME has been identified and consulted and concurs with the request as a valid measurement.
- The action (possible) that is taken in response to the reception is known.
- The actor responsible for the action is identified and is aware of the event and agrees to action it when notified (could be automated if possible.)

At this point, the monitor does not exist. Prior to the allocation of resources to develop the solution, all of these sign-offs must be complete. Failure to do so will likely result is dissatisfaction and the inability to recognize or achieve the value of the solution.

8.2.8 Create, test, and implement the monitor

First and foremost, separation of duties should be utilized: creator of the monitor is *not* the tester, implementer is not the tester, and so on. All of these parties should sign off on their responsibilities for this monitor.

Some small amount of work is likely to remain when the solution becomes ready for production and for final user acceptance. This is likely to be customization to fit final or late identification of needs. Do not forget to develop documentation and training for the users.

Important: A development environment is required to develop the monitors. A test environment is required to test the monitors. The way that the product functions, you are making changes in the production environment whenever you are working to create monitors if you are not working on an isolated system. Doing so invites problems at some point and is definitely *not* recommended.

The requirements for your test and development environments are not high and could be met with virtual machines. If necessary, test and development could be one and the same; however, it is not best practices to have them be the same. If they are different, it also gives a test to the import of them into another environment such as you are going to do in production.

Security in the production environment should be locked down to the point that only a very select group can make changes to situations running in production. Any changes that must happen in the production environment must be approved by the change control process or through an emergency change proviso.

Release management should also play a role in development. As you build your monitors, you should export your work and save it in a directory structure that enables you to version your work. That way, you will be able to restore the work to some previous point if you should accidentally introduce some problem at a future point to a known good save.

The facility that enables you to export the situations from your test environment to re-import later or to import into your production environment is a set of command-line commands:

```
tacmd viewSit {-s|--situation} SITNAME [{-m|--system} SYSTEM]
[{-e|--export} [FILENAME] ]]

-s|--situation: Specifies the name of the situation to view.
-m|--system: Specifies the managed system to view the situation definition
for.
-e|--export: Exports the situation definition to a file of the name
specified.
```

The re-import of the created XML file into the test environment later or into the production environment is done through the facility to re-import these XML files:

```
tacmd createSit {-i|--import} [FILENAME]

-i|--import: Specifies the situation definition to import.
```

In the next section of this chapter, we explore an example of this in detail.

8.3 Building the monitoring

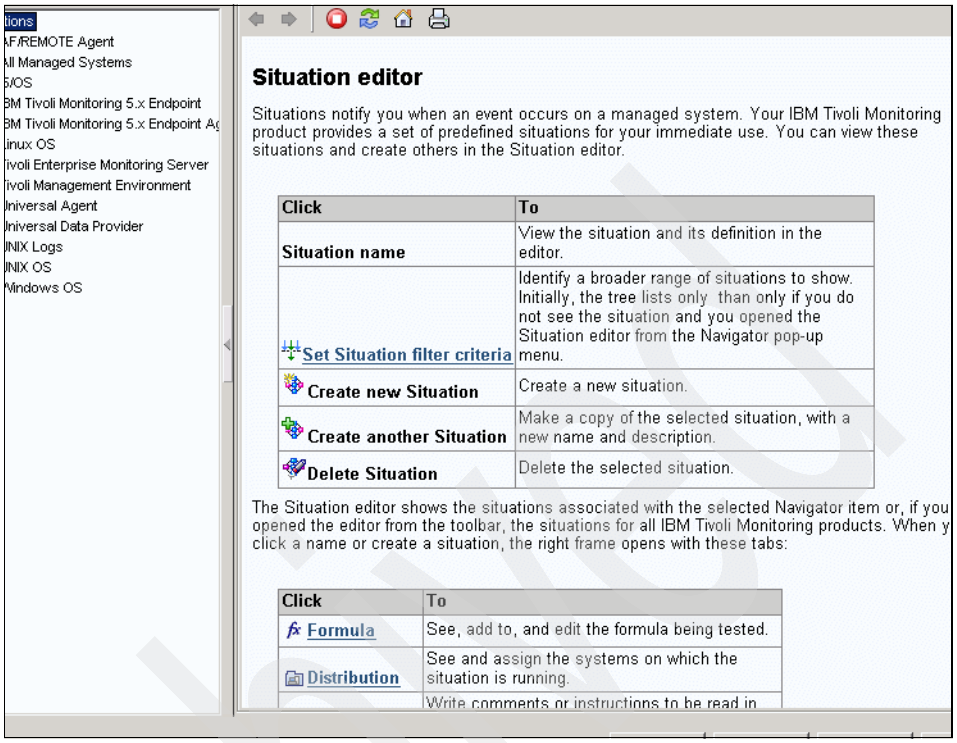
In building the monitoring within IBM Tivoli Monitoring 6.1, there is a choice between the situation editor and the command line. First we discuss building situation monitors in the editor and then the GUI, because when we look at the GUI, we can see the formula that we would be building with the command line. It will be easier to understand the options if we explore the situation editor first.

Building a complete situation involves the following considerations:

- Naming the situation
- Selecting attributes for comparison
- Editing the formula
- Selecting targets of distribution: MSL, systems, or both
- Writing expert advice to explain the situation event
- Setting the action
- Using the Until tab
- Best practices for situation creation

8.3.1 Naming the situation

The situation editor is launched from the Tivoli Enterprise Portal browser or desktop by issuing the key sequence Control-E. The Situation Editor window (Figure 8-2 on page 483) appears and provides a list of situations in your monitoring environment.


tions
F/REMOTE Agent
ll Managed Systems
/OS
BM Tivoli Monitoring 5.x Endpoint
BM Tivoli Monitoring 5.x Endpoint A
inux OS
ivoli Enterprise Monitoring Server
ivoli Management Environment
niversal Agent
niversal Data Provider
NIX Logs
NIX OS
Vindows OS

Situation editor

Situations notify you when an event occurs on a managed system. Your IBM Tivoli Monitoring product provides a set of predefined situations for your immediate use. You can view these situations and create others in the Situation editor.

Click	To
Situation name	View the situation and its definition in the editor.
Set Situation filter criteria	Identify a broader range of situations to show. Initially, the tree lists only than only if you do not see the situation and you opened the Situation editor from the Navigator pop-up menu.
Create new Situation	Create a new situation.
Create another Situation	Make a copy of the selected situation, with a new name and description.
Delete Situation	Delete the selected situation.

The Situation editor shows the situations associated with the selected Navigator item or, if you opened the editor from the toolbar, the situations for all IBM Tivoli Monitoring products. When y click a name or create a situation, the right frame opens with these tabs:

Click	To
Formula	See, add to, and edit the formula being tested.
Distribution	See and assign the systems on which the situation is running.
	Write comments or instructions to be read in

Figure 8-2 The situation editor

You should take a few moments to explore the types of situations that exist out of the box.

Tip: It is possible to get to the situation editor in several ways from the Navigator other than our choice of Ctrl + E. If you right-click the Enterprise item and select the situation editor icon, you see the situations that were distributed to the monitoring server. If you right-click a system item to see the situation editor icon, you see all situations that were distributed to that managed system. If you right-click an agent item (or if the agent has subagents), you see all situations that have been distributed to the monitoring agent. At the attribute level, you can see situations distributed to that managed system but only those that were created with related attribute groups.

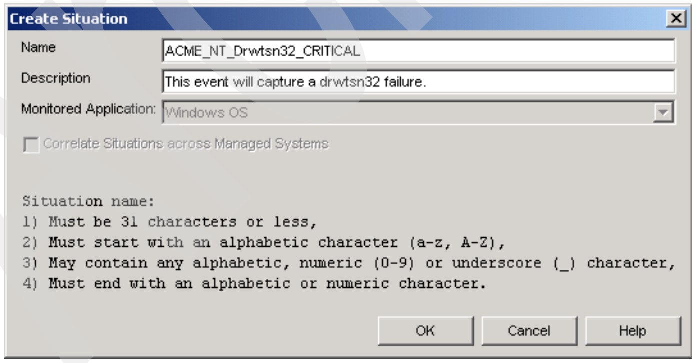
With Ctrl + E, we see the realm of options in the environment. We are able to build situations that are not associated with a Navigator icon by doing this.

In our first scenario, we have been charged with coming up with an alerting mechanism that a drwtsn32 alert has been generated on a Windows system. A mission-critical application for our environment has been found to be suffering from some problem that results in this failure. Although we would prefer to have the application programmers create a more proactive way to alert us to the problem, this is the only existing way that the situation can be recognized (in our scenario). Additionally, we understand that some other key applications might fail and generate these events. We will refine this concept as we move through this discussion to add new features to this monitoring solution.

The first step is to name the situation (Figure 8-3). We consider that the Windows OS agent will be used to make the monitoring happen. Therefore, we must click the plus icon to expand the Windows OS category. We then click the Windows OS title for the category and right-click the icon above to create a new situation.

Because in our environment we wish to generate an IBM Tivoli Enterprise Console event, we name the situation with _CRITICAL on the end of the name. This works with the IBM Tivoli Enterprise Console integration of IBM Tivoli Monitoring 6.1 to automatically set the severity of the event to critical. Our other options include _WARNING or to manually map the criticality of the event in the mapping file. See Chapter 10, “Integration with IBM Tivoli Enterprise Console” on page 583 for more information. We might choose Drwtsn32_CRITICAL.

For our first iteration of this situation, we merely look for the existence of a process. When a Dr. Watson alert is caused, a pop-up appears on the screen and an entry (multiple-line) is made in the log. While the pop-up is present on the screen, a process appears in the process table that we can identify. The method that we choose for identifying the problem is to look for this process. Obviously, there are some flaws in the methodology but we use it for the purpose of this example. We will continue to refine our sophistication throughout the examples.



Name	ACME_NT_Drwtsn32_CRITICAL
Description	This event will capture a drwtsn32 failure.
Monitored Application:	Windows OS

☐ Correlate Situations across Managed Systems

```
Situation name:
1) Must be 31 characters or less,
2) Must start with an alphabetic character (a-z, A-Z),
3) May contain any alphabetic, numeric (0-9) or underscore (_) character,
4) Must end with an alphabetic or numeric character.
```

Figure 8-3 Naming the situation

When we name the situation, certain practices will be useful. First, look at the situation names that are defaulted with the product. Note that the NT alerts begin with NT_. This can be good. However, it is helpful to make all of your situations stand out by naming them with the name of your company at the beginning.

An example format for a standard is:

```
<Customer>_<Component>_<Description>_<Severity>
```

Revising our name under this format, if our company was Acme and the component we are interested in is the NT component, we might name it ACME_NT_Drwtsn32_CRITICAL. The Description field that you populate in this dialog will populate the fly-over text of the situation in the navigator.

You might also name the situation with a responsibility group in the name. SYSADMIN_NT_Drwtsn32_CRITICAL might be a choice, as might NASYSADMIN_... (North America) or EURSYSADMIN_... (Europe) for other examples. The reason for doing this might be that your organization cannot standardize on situations (highly recommended as one size can fit all if you can make them agree.) Naming them with the same prefix will make them appear in the situation navigator in alphabetical order grouping as well.

The example format might be:

```
<Responsibility>_<Component>_<Description>_<Severity>
```

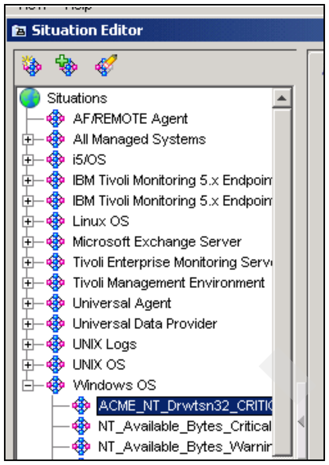
You are limited to 31 characters for situation names, so you might have to create abbreviations. For example, instead of PRICE_WATERHOUSE_COOPERS_NT, you should choose PWC_NT. Also, the Tivoli Enterprise Console integration _CRITICAL and _WARNING can be abbreviated to _CRIT and _WARN if necessary to meet the 31-character limit.

Tip: One product limitation is that if you save the name by closing this dialog box, you will be forced to copy the situation to rename it. The function of duplicating a situation (and other parts of this GUI use a similar approach) is called *Create Another*. It is not particularly intuitive that this is what this menu option is intended to do, but it is essentially Create a Duplicate Copy. The first thing it does in the situation editor is provide you with a dialog box to name the copy (and thus you can rename). You will then have to delete the version with the name you wanted to change.

This is a particularly good reason to develop your naming standards before you ever write the first monitor situation. If you do not, you will be forced to create duplicates for all existing custom situations in order to rename them, then forced to delete all the incorrectly named copies. This can involve a lot of work for someone later when your standard is created.

8.3.2 Selecting attributes for comparison

After we have finished naming our situation, we select attributes for comparison in our situation. Figure 8-4 shows that the named situation has appeared in the list of situations. We click **Add conditions** to gain access to this list of attributes.

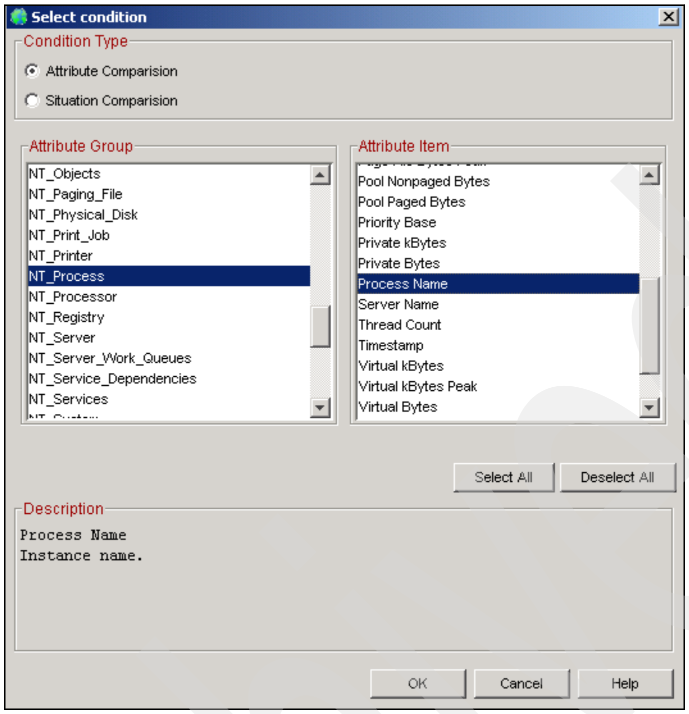


Situation Editor
Situations
AF/REMOTE Agent
All Managed Systems
i5/OS
IBM Tivoli Monitoring 5.x Endpoin
IBM Tivoli Monitoring 5.x Endpoin
Linux OS
Microsoft Exchange Server
Tivoli Enterprise Monitoring Serv
Tivoli Management Environment
Universal Agent
Universal Data Provider
UNIX Logs
UNIX OS
Windows OS
ACME_NT_Drwtsn32_CRITI
NT_Available_Bytes_Critical
NT_Available_Bytes_Warnir

Figure 8-4 Our situation name appears in the situation editor list

Note: If the name of the situation is incorrect or it is located in the wrong place in the situation list, delete it and create it again in the correct location.

We have previously discussed that we will look for a process name in the process table. Therefore, we select the attribute group for NT_Process and the individual attribute as Process Name (Figure 8-5 on page 487). For the purposes of our simple example, this is sufficient.



Condition Type: (●) Attribute Comparision ( ) Situation Comparision

Attribute Group: NT_Process

Attribute Item: Process Name

Description

```
Process Name
Instance name.
```

Figure 8-5 Attribute comparison selection

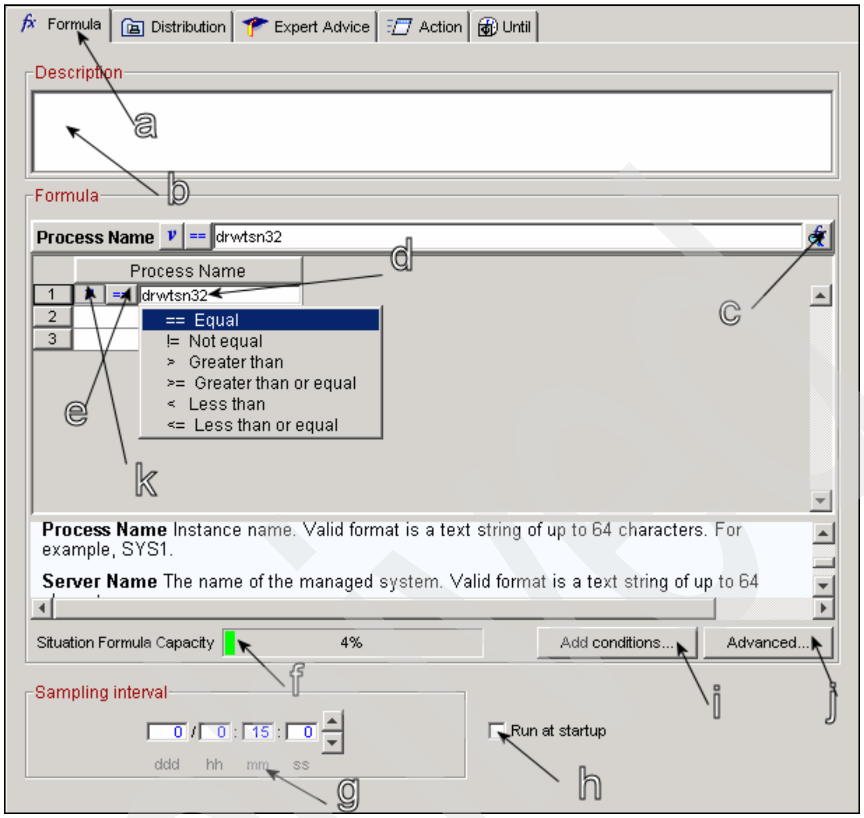
When we click **OK**, the editing window appears.

8.3.3 Editing the formula

Important: You are limited in the number of attributes that you can select from this list by the size of the formula that is created with the selection and creation of criteria for those attributes. The number of attributes that you can have used to be a hard stop of 10 in the OMAGAMON product. That restriction has been lifted and replaced with a status bar that indicates the current size of the formula against the total size possible. The current formula limit is 1020 characters.

It is still recommended, however, that if you have multiple situations using the same attribute group, that you limit the number of attributes to 10 for performance reasons.

Figure 8-6 on page 488 shows the formula editing window. We will refer to the labelled points in the figure to explain what you will need to know.



Process Name Instance name. Valid format is a text string of up to 64 characters. For example, SYS1.

Server Name The name of the managed system. Valid format is a text string of up to 64

Figure 8-6 The formula editing window

a. Note that the formula tab is selected.

b. Type a short description here. You are limited to 31 characters.

c. This button will show the completed formula.

d. In this cell we type the name of the process we are going to match.

e. This button brings up the equality choices list (not equal, equal, greater than, and so on)

f. This shows the formula capacity. The total limit of characters is 1020 for the formula and this bar shows your percentage of that in the formula. Refer to button “c”.

g. The sampling interval is set here. By default it is 15 minutes. See the previous discussion about considerations in item 2 on page 476.

h. Run at startup is checked by default. If this box is checked, as soon as you save the situation it will start to run to all subscribed systems.

i. Add conditions. We have already used this. This brings up our attribute selection dialog.

j. Advanced. This brings up two more options. One is persistence. The other is called display item.

k. Functions. The list of available functions varies by the type of attribute. The default is value of expression. For a complete list of choices, refer to the online help or *IBM Tivoli Monitoring, Version 6.1.0, User's Guide,* SC32-9409.

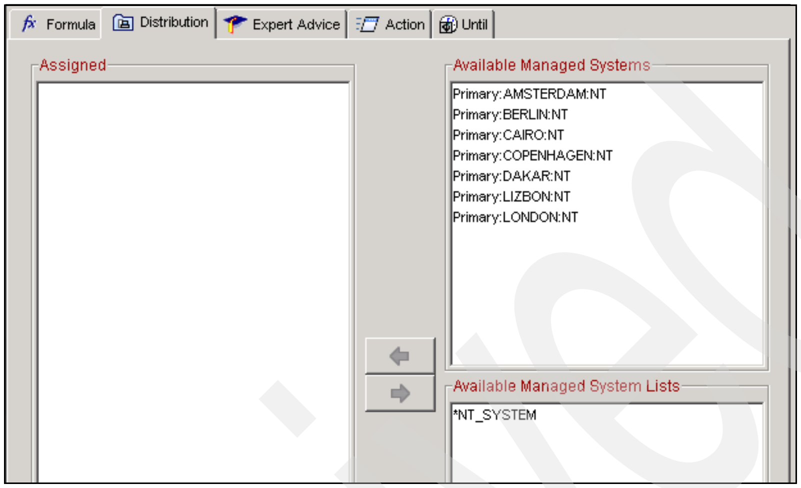
Important: The persistence setting requires that you wait a selected number of occurrences of TRUE evaluation prior to creating an event. In IBM Tivoli Monitoring 5.x terms, this is a choice of the number of occurrences; however, you are limited to zero holes. This means that you cannot be as sensitive to outliers. (If it happens nine times over the course of 10 sample cycles and the situation is measured as false on the 10th measurement, it would not alert you unless there are 10 in a row.)

The display item choice opens a dialog that lets you pick from only certain attributes to continue checking for in the sampling and create other alerts. If it were possible, for instance, to have multiple Dr. Watson pop-ups on the screen (and thus multiple processes in the process table), we could use this dialog to cause separate events for each one on the system. The default behavior would only give you a single event in this scenario.

For our example, we have the process name drwtsn32, set our sampling interval to 30 seconds, and set it to run at startup.

8.3.4 Selecting targets of distribution: MSL, systems, or both

Next we look at the Distribution tab for our options. Figure 8-7 shows the Distribution tab.



Primary:AMSTERDAM:NT
Primary:BERLIN:NT
Primary:CAIRO:NT
Primary:COPENHAGEN:NT
Primary:DAKAR:NT
Primary:LIZBON:NT
Primary:LONDON:NT

*NT_SYSTEM

Figure 8-7 The Distribution tab dialog

The Distribution tab offers the choice of single systems or managed system lists. These managed system lists are groups of servers and are defined by the edit button. The MSLs that are predefined for you show up with an asterisk at the beginning of their name. While the systems automatically go into platform managed systems lists by platform, you will have to move the systems into MSLs manually.

In our example, we pick the MSL for *NT_SYSTEM so that all new NT systems added to the environment get this situation by default.

8.3.5 Writing expert advice to explain the situation event

The next tab in the Situation Editor is the Expert Advice tab (Figure 8-8).

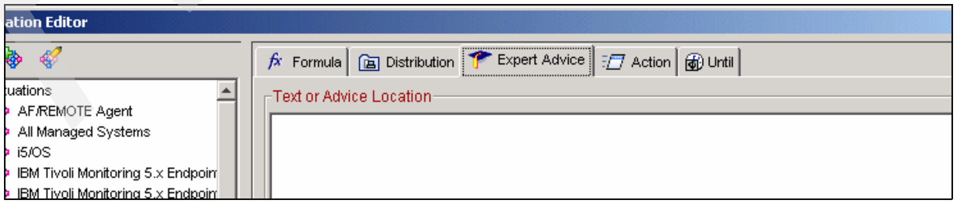


Figure 8-8 The expert advice tab.

You can write text in this expert advice tab and it will display in the expert advice pane as text.

You can also write HTML code in this space. It will be handled appropriately. In this space, you can use the variable expression syntax exactly as you can in the link wizard, including input such as Example 8-1.

Example 8-1 Variable expression example

```
situation name           $EVENT:ATTRIBUTE.ISITSTSH.SITNAME$
monitoring server name   $EVENT:ATTRIBUTE.ISITSTSH.NODE$
managed system name      $EVENT:ATTRIBUTE.ISITSTSH.ORIGINNODE$
display item             $EVENT:ATTRIBUTE.ISITSTSH.ATOMIZE$
global timestamp         $EVENT:ATTRIBUTE.ISITSTSH.GBLTMSTMP$
local timestamp          $EVENT:ATTRIBUTE.ISITSTSH.LCLTMSTMP$
status                   $EVENT:ATTRIBUTE.ISITSTSH.DELTASTAT$
```

If you wish to redirect the user to an existing Web page, simply put the URL in the pane without any other text:

```
http://www.cisco.com/warp/public/78/drwtsn32.shtml
```

The browser will appear and launch to this page. You also can use variables if you wish to launch the Web page in the context of some variable, as in passing information to a search engine.

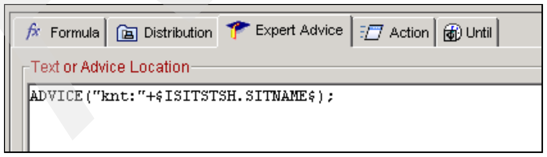
Important: Due to the limitation of 512 characters, you will likely want to use an external Web file as the product does for the built-in situation expert advice.

This does not allow you to use the variable substitution. If you wish to use variable substitution, you must stay within the 512-character limit.

This procedure could be used to build expert advice using an external HTML file:

1. In the Expert Advice box, enter the same information that exists in one of the default situations (Figure 8-9):

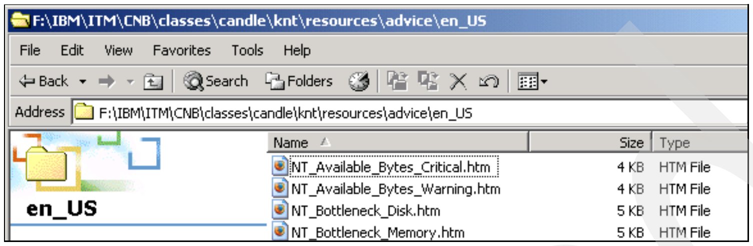
```
ADVICE("knt:"+$ISITSTSH.SITNAME$);
```



```
ADVICE("knt:"+$ISITSTSH.SITNAME$);
```

Figure 8-9 Populate the expert advice tab like this

2. Copy one of the IBM Tivoli Monitoring 6.1 expert advice HTML files (Figure 8-10) and create one with the name matching your situation. In our case, we create `ACME_NT_Drwtsn32_CRITICAL.htm`. It should be placed in the same directory as the other HTML files.

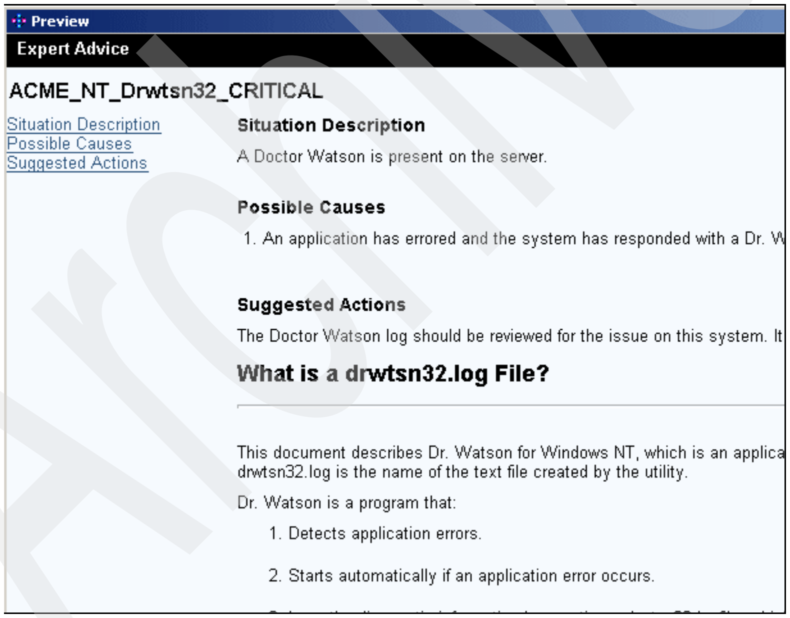

F:\IBM\ITM\CNB\classes\candle\knt\resources\advice\en_US

File Edit View Favorites Tools Help
Back Search Folders
Address F:\IBM\ITM\CNB\classes\candle\knt\resources\advice\en_US
en_US

Name	Size	Type
NT_Available_Bytes_Critical.htm	4 KB	HTM File
NT_Available_Bytes_Warning.htm	4 KB	HTM File
NT_Bottleneck_Disk.htm	5 KB	HTM File
NT_Bottleneck_Memory.htm	5 KB	HTM File

Figure 8-10 Default location of expert advice files on Windows TEPS server

3. Edit the expert advice file to contain your desired expert advice information. Save the HTML file.
4. Click **Preview** to see how it looks (Figure 8-11).



ACME_NT_Drwtsn32_CRITICAL

Situation Description

A Doctor Watson is present on the server.

Possible Causes

1. An application has errored and the system has responded with a Dr. W

Suggested Actions

The Doctor Watson log should be reviewed for the issue on this system. It

What is a drwtsn32.log File?

This document describes Dr. Watson for Windows NT, which is an applica
drwtsn32.log is the name of the text file created by the utility.

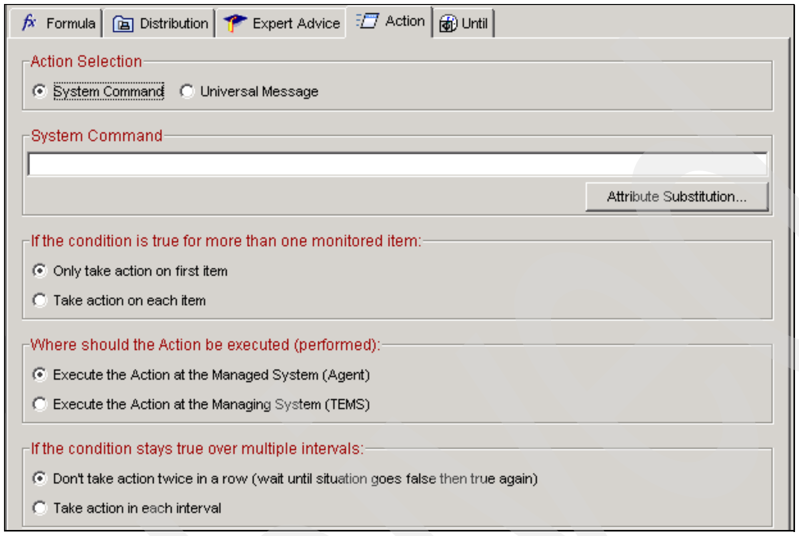
Dr. Watson is a program that:

1. Detects application errors.
2. Starts automatically if an application error occurs.

Figure 8-11 Our custom expert advice for the ACME event

8.3.6 Setting the action

Next, set the actions for when the situation becomes true. The other options in the action tab are described following Figure 8-12.



Action Selection
[✓ System Command] Universal Message

System Command

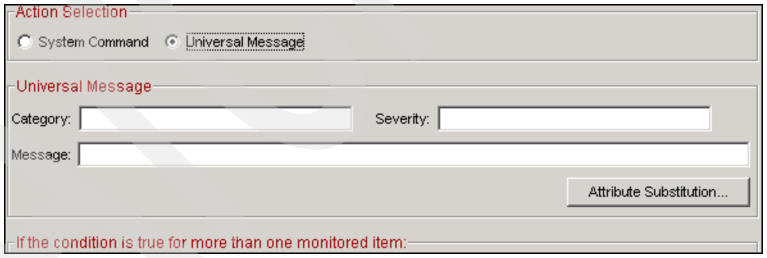
If the condition is true for more than one monitored item:
[✓ Only take action on first item]
Take action on each item

Where should the Action be executed (performed):
[✓ Execute the Action at the Managed System (Agent)]
Execute the Action at the Managing System (TEMS)

If the condition stays true over multiple intervals:
[✓ Don't take action twice in a row (wait until situation goes false then true again)]
Take action in each interval

Figure 8-12 The Action tab

The Action tab offers two main options: system command and universal message. Figure 8-12 shows the system command choice. Figure 8-13 shows the changed part of the screen if universal message is selected.



Action Selection
System Command [✓ Universal Message]

Universal Message
Category:
Severity:
Message:

If the condition is true for more than one monitored item:

Figure 8-13 Action tab with universal message selected (the differences only)

Under the system command selection, any system command can be issued. There is a button for attribute substitution that enters a variable so that when the situation is true, appropriate values will be used.

As an example, the system command might be something like one of these:

```
net stop oserv
echo "There is a serious error on the system" | mail root
```

Of course, the system command must be platform appropriate. An example of using the variable substitution is to have the process name be generic and thus have a more multi-purpose situation. Our next example explores this further.

Other options on the Action tab enable you to customize the response, such as:

- Take action on the first item.
- Take action on every item.

The differentiation is where possibly there could be multiple matches to the same attribute condition check (thus returning a result of multiple rows of data). The selection of Take action on every item will cause your system command to be executed for each line. We also look at this in our next situation.

The next group of actions on the tab are to select where the action is issued: on the agent machine or the TEMS machine. This might be useful if, for example, you want to build a log file of these entries in a single file on the TEMS server:

```
echo " &NT_Process.Timestamp There has been a Dr Watson on
&NT_Process.Server_Name" >> /logs/DrWatsonRollup.log
```

The final section controls how many times you might run the system command; that is, whether it issues the command twice in a row or waits for the situation to evaluate false again (symptom gone) or if it should issue the command at every sample that is true.

Important: Pay attention to sample interval here (as previously discussed). For example, if you know that a command takes more than 30 seconds to complete, it does not make sense to set your sample interval to 15 seconds and then pick fire every time, because it will issue the command at least two times before the first attempt has a chance to be successful.

Under the Universal Messages selection, you define a universal message for the universal message log. You define the category the message should fall under, such as Non-essential or SiteCritical. Type a one-word term of up to 16 characters to categorize the message. Severity is the importance of the message, such as High or 1. Type a one-word term of up to eight characters that conveys the severity of the message. Message is the text to display in the UML when the situation occurs. Type a message of up to 245 characters. You can include an attribute value in the message by selecting Attribute Substitution.

8.3.7 Using the Until tab

The Until tab contains two separate functions for a single purpose. The purpose of this tab is to designate when to close the event automatically. If you never want it to be closed automatically, do not choose anything on this tab.

The first function is to close the event when another event also becomes true. You are limited to the same agent type and the same set of attributes. The second function is to close the event at some time interval.

Important: Do *not* use the Until tab for situations that are sampled. The sampled events, as you may recall, should not be manually closed. The result of having them close is that the situation goes to a status of closed. It is not until the situation would recover (become false) and then suffer again (become true) that you will see this event again.

As an example: We set the NT_Missing_Process to look for Notepad on our system. Then we set an until for 30 seconds. With Notepad not running, in 30 seconds, the event closes and soon after disappears. Although our "critical" situation is still technically existing, the tool cannot alert us until somebody starts Notepad again and then closes it.

8.3.8 Best practices for situation creation

Data for situations and events is collected at regular intervals. However, situations often do not need to be active on a 24 x 7 basis. For example, many alerts may only be required during normal business hours. The first way to control resource usage by situations is to stop and start them at the times that they are required. This can be accomplished by creating policies (that start and stop situations at the right time) or externally, by using an automation or scheduling software that starts and stops situations using Web services.

Discussing with end user departments which situations should be built and how is one of the most critical success factors of the project. If critical components are not being monitored by a situation or not by using the right thresholds, problems risk arising without being noticed.

If the situation intervals are set too short, the overhead will be too high and the overall implementation may even become unreliable when the components cannot handle the workload any more; for example, if the TEMS is evaluating more situations than it can handle, or receiving more alerts, it will start to queue them, generating even more overhead and delaying critical alerts to be raised.

If the situation interval is too long, problems may be detected too late. Hence the importance of in-depth planning and review with the end user department and the need for the department to delegate a senior member to assist with this project. If the end user representative is a junior member, he or she may not be sufficiently aware of critical performance factors and may even lack sufficient authority to defend the outcome of the discussions with his department.

The following items should be discussed:

- Critical performance factors for the application or system. These must be translated into data attributes to be monitored by using them in situations.
- What are the best values to check these attributes against? Should multiple situations be created to watch several levels of severity?
- If you do need several levels of severity for the same data, keep the sampling interval the same: they will be grouped together and data will only be collected once (see grouping situations below).
- Select realistic alert values: For example, if a situation triggers and resets frequently, then the TEP user in Operations is over asked and may lose reactivity over time. Moreover, this causes unnecessary overhead to the TEMS: Processing is required to handle the alerts, but also to store the data that leads to the alert.
- At which intervals should these factors be checked? Also check whether the data that is required for the situation is not collected in background: some TEMA data (mainly Plex mainframe data) is not collected on demand, but rather at fixed interval. These data collectors are running as UADVISOR probes in the TEMA/TEMS. The situation that uses these data should have an interval of at least the UADVISOR interval; otherwise, the same UADVISOR collected data will be used twice or more for situation evaluation.
- Which systems should be monitored? Systems should be grouped into user-managed system lists. Situations will then be distributed to these MSLs. When a system should be added for the same kind of alerting afterward, the only change required will be to the MSL.
- When the situation triggers, what advice can be give to the operator? This information will be put into the situation advice and will be presented to the operator when the alert is raised and advice is selected. This way the operator is assisted to take the right actions — action that is consistent with the company's policies.
- Is any automated action required? And if so, what? This will result in either a simple command to be executed on the system (reflex automation in the situation) or in a more complex set of automation scripts (that will be added into a policy).

Situations grouping

Grouping situations can potentially save a lot of resources, but can unfortunately not be set manually; the TEMS will decide whether to group situations. The following conditions must be met before a situation can be part of a group:

- All situations in the group should use elements from the same attribute group.
- Must use the same interval setting.
- Must have autostart YES.
- Cannot contain an UNTIL clause.
- Distribution lists may be different.
- Cannot contain a display item.
- Cannot contain a take action item.
- MISSING function is not supported.
- SCAN and STR functions are not supported.
- Group functions on the attribute criteria (such as average or total) are not supported.
- Event persistence is not supported.

If the situation is grouped with other situations, the data collection required to get the attributes that are referenced in the situation will be done only once for the group. All situations in the group will make use of the same data.

Situation grouping is done by TEMS during its start-up: If the TEMS finds a number of situations that are eligible for grouping, it will create a new internal situation that will perform the data collection at the specified interval.

All grouped situations will then compare their criteria to the data returned by the internal situation. These internal situations only exist for the duration of the TEMS run. They get an internal name that starts with _Z_ and the full name will be built from the following parts: _Z_, table name, sequence number.

For example, on Windows, when grouping situations on table WTPROCESS, the grouped situation will be called _Z_WTPROCESS0. These situations are not added to the permanent situation tables in TEMS (such as TSITDESC), but since they are only temporary, they can only be seen in situation temporary tables such as TSITSTSC.

To verify whether any grouped situations are created, you can run a SQL statement from a TEP View, using custom SQL (Example 8-2 on page 498).

Example 8-2 SQL statement from a TEP View

```
SELECT SITNAME,
       ATOMIZE,
       DELTASTAT,
       LCLTMSTMP,
       NODE,
       ORIGINNODE,
       RESULTS,
       SITCOUNT,
       TYPE
FROM O4SRV.TSITSTSC;
```

The grouping occurs only at TEMS start-up, so any new situations or modifications will not benefit from grouping until the TEMS is restarted.

Where is the situation evaluated?

Situations can be evaluated at either TEMA or TEMS. Ideally, all situations would evaluate at the TEMA: as close to the data source as possible. Unfortunately, the TEMA is limited in its capacities to evaluate the situation; the evaluation will be moved to the TEMS in the following conditions:

- If the situation has attributes that cross TEMAs
- If advanced checking is used (such as string scan)

If situations cannot be evaluated at the TEMA, the TEMS will take over. You should avoid evaluating situations at the HUB TEMS; all TEMAs should report to a Remote TEMS.

Building a situation in the right order

When starting to build a new situation, you should first make an overview of the attributes to test. Attributes will be tested from first to last, or from left to right on the TEP panel, in the order they are entered in the situation.

Knowing the data behind attributes is recommended. The first test to make should return as few rows as possible; the next step can then further filter a limited set of rows. For example, on Windows, to check whether process XYZ uses more than n amount of real storage, we have to test two attributes (process name and real storage usage).

If we first test on real storage use, the result set may contain multiple rows; then we check whether our process name is among the returned rows. It would be more efficient to first test on process name (the result will be one row), followed by the test on the storage usage, just on this single row.

Also, when using complex conditions, such as string scan, sum, or average, this can best be performed on a limited result set: First evaluate the attributes against simple conditions to reduce the result set.

8.3.9 Release management

After you have completed defining the situation and testing it to your satisfaction, you are ready to put it into your versioning control system. As previously mentioned, we will use the **tacmd** command-line utility to export the situation.

Attention: Note that if you issue the command without -e and simply redirect to a file you are not exporting the XML. You are exporting a summary text to a file. You must use the -e option to obtain the correct output.

First you are required to authenticate. Then you issue the **tacmd** to export the file. It is helpful to give it a version number. You should move this file to the repository of your monitoring definitions for backup in the event of a disaster as well as in the event that changes are made to the situation that are later found to be undesirable and a previous state of the situation is more desirable.

Example 8-3 Exporting a situation to an XML file

```
[root@milan][/opt/IBM/ITM]-> tacmd login -s milan -u sysadmin

 Password?

Validating user...

KUIC00007I: User sysadmin logged into server on https://milan:3661.
[root@milan][/opt/IBM/ITM]-> tacmd viewsit -s ACME_NT_Drwtsn32_CRITICAL -e
/tmp/ACME_NT_Drwtsn32_CRITICAL_v1_0.xml

KUICVS004I: The Situation ACME_NT_Drwtsn32_CRITICAL was exported to
/tmp/ACME_NT_Drwtsn32_CRITICAL_v1_0.xml.
```

8.4 Building another example: monitoring Tivoli

For the purpose of having another real-life scenario, we explore monitoring the Tivoli infrastructure. In this example we build some components to monitor our Tivoli environment.

We look purely at the availability of the Tivoli Framework. Note that situations are platform specific and you will have to build similar situations for Windows and UNIX that will differ in the attributes used.

The monitors that we build in this section include:

- 8.4.1, “Monitoring the UNIX oserv” on page 500
- 8.4.3, “Monitoring the Tivoli UNIX gateways” on page 509

8.4.1 Monitoring the UNIX oserv

We begin with monitoring for the oserv on our UNIX systems because our TMR is a UNIX system. We copy the UNIX situation UNIX_CMD_Process_Critical and rename it to ACME_UNIX_oservavail_CRITICAL.

We look for the missing process of the oserv and create some expert advice. Our sample interval is set to 1 minute in this situation. We will set the action to mail some text to a Sprintpcs account for paging.

Example 8-4 shows the text export of this situation and the XML file produced.

Example 8-4 ACME_UNIX_oservavail_CRITICAL situation

```
[root@milan][/opt/IBM/ITM]-> tacmd login -s milan -u sysadmin

 Password?

Validating user...

KUIC00007I: User sysadmin logged into server on https://milan:3661.

[root@milan][/opt/IBM/ITM]-> tacmd viewsit -s ACME_UNIX_OservAvail_CRITICAL
Name                      : ACME_UNIX_OservAvail_CRITICAL
Description               : checks for running oserv
Type                      : UNIX OS
Formula                   : *IF *MISSING Process.Command *EQ ('*oserv *')
Sampling Interval         : 0/0:1:0
Run At Start Up           : Yes
Distribution              : istanbul.itsc.austin.ibm.com:KUX
Text                      : The oserv on $EVENT:ATTRIBUTE.ISITSTSH.ORIGINNODE$
is down. The Tivoli on-call person has been paged.
Action Location           : Agent
Action Selection          : System Command
Universal Message Category:
Universal Message Severity:
Universal Message         :
True For Multiple Items   : Action on First item only

[root@milan][/opt/IBM/ITM]-> tacmd viewsit -s ACME_UNIX_OservAvail_CRITICAL -e
/tmp/ACME_UNIX_OservAvail_CRITICAL_v1_0.xml
```

```
KUICVS004I: The Situation ACME_UNIX_OservAvail_CRITICAL was exported to
/tmp/ACME_UNIX_OservAvail_CRITICAL_v1_0.xml.

[root@milan][/opt/IBM/ITM]-> cat /tmp/ACME_UNIX_OservAvail_CRITICAL_v1_0.xml

<TABLE>
<ROW>
<SITNAME>
ACME_UNIX_OservAvail_CRITICAL
</SITNAME>
<TEXT>
<![CDATA[checks for running oserv]]>
</TEXT>
<AFFINITIES>
00f2000000000000000000000000000##00000w0a7
</AFFINITIES>
<PDT>
<![CDATA[*IF *MISSING Process.Command *EQ ('*oserv *')]]>
</PDT>
<REEV_DAYS>
0
</REEV_DAYS>
<REEV_TIME>
000100
</REEV_TIME>
<AUTOSTART>
*YES
</AUTOSTART>
<ADVISE>
<![CDATA[The oserv on $EVENT:ATTRIBUTE.ISITSTSH.ORIGINNODE$ is down. The Tivoli
on-call person has been paged.]]>
</ADVISE>
<CMD>
<![CDATA[echo "oserv on $EVENT:ATTRIBUTE.ISITSTSH.ORIGINNODE$ down" | mail
tivoncall@sprintpcs.com]]>
</CMD>
<AUTOSOPT>
NNN
</AUTOSOPT>
<DISTRIBUTION>
istanbul.itsc.austin.ibm.com:KUX
</DISTRIBUTION>
<ALERTLIST>
</ALERTLIST>
<HUB>
</HUB>
<QIBSCOPE>
E
</QIBSCOPE>
```

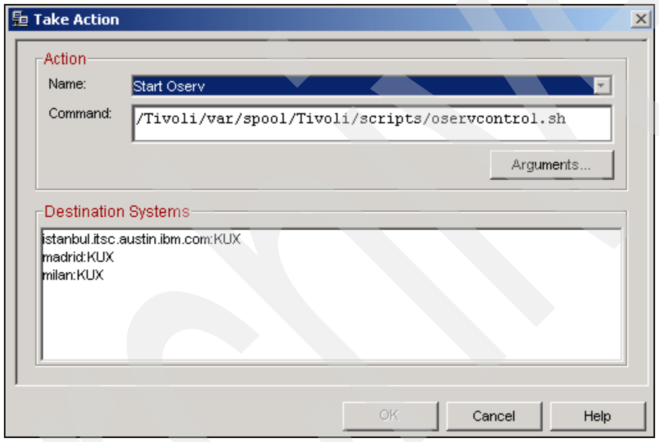
```
<SENDMSGQ>
*NONE
</SENDMSGQ>
<DESTNODE>
</DESTNODE>
<LOCFLAG>
</LOCFLAG>
<LSTCCSID>
en_US
</LSTCCSID>
<LSTDATE>
1051021135745000
</LSTDATE>
<LSTRELEASE>
V100
</LSTRELEASE>
<LSTUSRPRF>
SYSADMIN
</LSTUSRPRF>
<NOTIFYARGS>
</NOTIFYARGS>
<NOTIFYOPTS>
</NOTIFYOPTS>
<OBJECTLOCK>
</OBJECTLOCK>
<PRNAMES>
</PRNAMES>
<REFLEXOK>
</REFLEXOK>
<SITINFO>
<![CDATA[~]]>
</SITINFO>
<SOURCE>
</SOURCE>
</ROW>
</TABLE>
```

When we explore the thoughts around monitoring the oserv, it is logical to assume that you might want to provide the facilities for the console operator to try to start the oserv before they try to call you. In this case, we rely on change management to notify the console operators if the stopping of the oserv was intentional. To facilitate the starting of the oserv, you can create a take action in the GUI. This is done through the take action facility (similar to the Action tab.)

The following picture shows how this could be done. First, as with building the situation itself, click the Navigator tab for the UNIX OS under one of the UNIX systems in your environment. Then right-click and select **Take action** → **Create/edit**.

This opens a dialog enabling you to configure the take action you want to create. In our example, we created a directory on the managed node in the install tree for Tivoli named scripts and wrote a script that sources the Tivoli environment and then issues the UNIX **`oserv`** command to start the oserv. This script can then be called by creating a take action when the operator right-clicks on the event in the event console.

If the script is successful at starting the oserv, the event should disappear at the next sample time, and when the on-call person calls in, the operator can report that the situation is resolved. A take action (Figure 8-14) can also be written to send a second page that the oserv start was successful. The operator could execute this when the event closes.



Name: Start Oserv

Command: /Tivoli/var/spool/Tivoli/scripts/oservcontrol.sh

```
istanbul.itsc.austin.ibm.com:KUX
madrid:KUX
milan:KUX
```

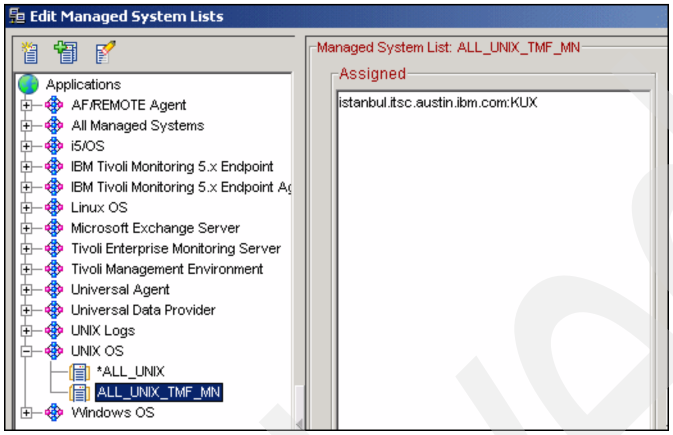
Figure 8-14   When take action is completed, operator will be able to reference it

Another alternative approach that we might use (not necessarily in this case) is to build a policy. We explore this in the next section.

After building a situation that can be distributed to all UNIX managed node systems to monitor the oserv itself, we would build a UNIX managed system list for all managed nodes in the Tivoli environment. This is a manual process in IBM Tivoli Monitoring 6.1.

To build a managed system list, press Ctrl + M to open the managed system list editor.

Because we want this to be associated with the UNIX OS agent, we select **UNIX OS** on the left and click the button to create a new managed system list. In our example, we add our TMR server (istanbul) to this list as shown in Figure 8-15.


Edit Managed System Lists
Applications
AF/REMOTE Agent
All Managed Systems
i5/OS
IBM Tivoli Monitoring 5.x Endpoint
IBM Tivoli Monitoring 5.x Endpoint Ag
Linux OS
Microsoft Exchange Server
Tivoli Enterprise Monitoring Server
Tivoli Management Environment
Universal Agent
Universal Data Provider
UNIX Logs
UNIX OS
*ALL_UNIX
ALL_UNIX_TMF_MN
Windows OS
Managed System List: ALL_UNIX_TMF_MN
Assigned

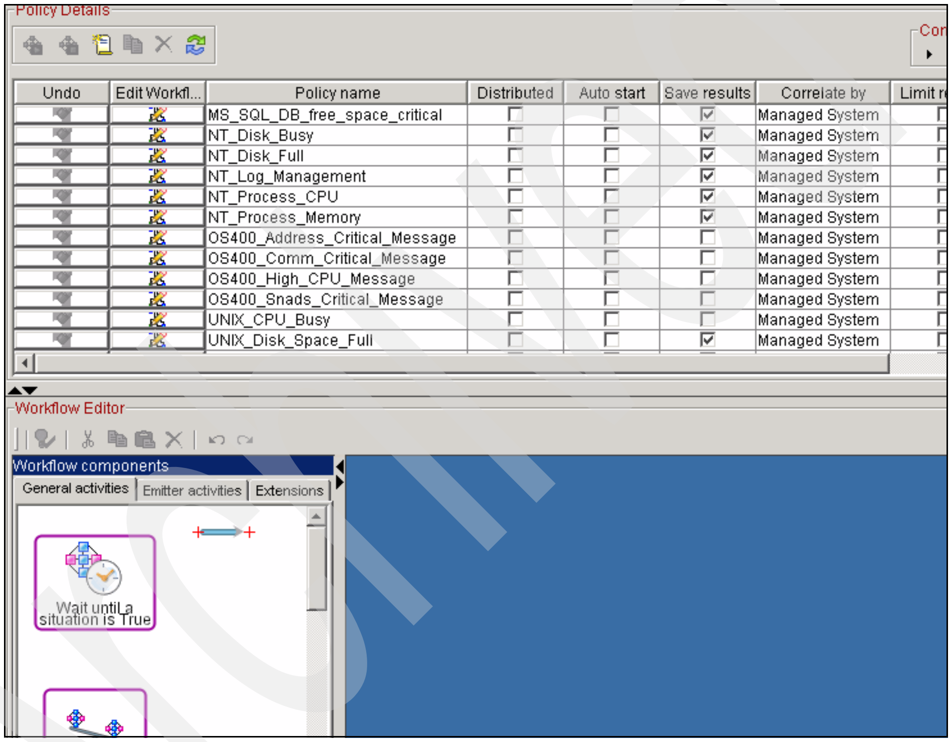
istanbul.itsc.austin.ibm.com:KUX

Figure 8-15 Setting istanbul TMR to be a member of the managed system list

8.4.2 Using a policy and workflow

The policy that we build is based on the idea that we want our monitor that detects the oserv to be down to attempt to resolve the issue and immediately alert us to the result, not wait for the five minutes to pass (in our example) before the sample occurs again to determine that the problem is resolved.

1. The first step is to create the workflow by accessing the Policy Editor with the TEP icon on the toolbar or pressing Ctrl + w (Figure 8-16). Click the New Policy button.

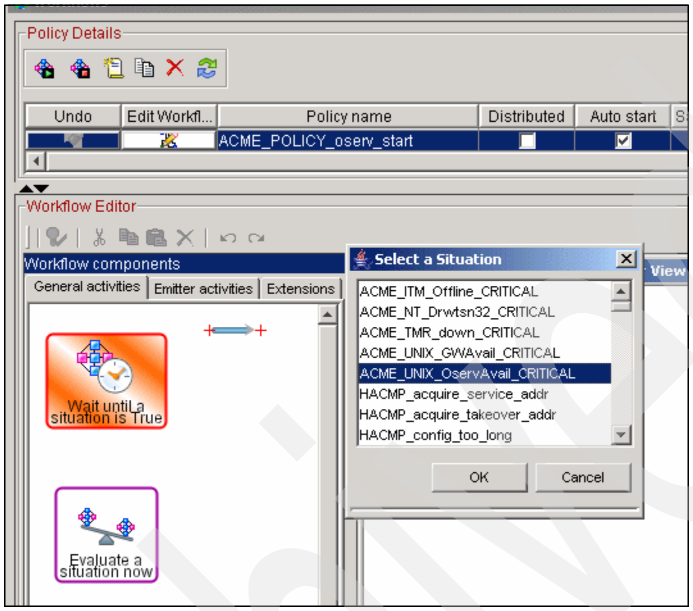


Undo	Edit Workfl...	Policy name	Distributed	Auto start	Save results	Correlate by	Limit r
		MS_SQL_DB_free_space_critical	☐	☐	✓	Managed System	☐
		NT_Disk_Busy	☐	☐	✓	Managed System	☐
		NT_Disk_Full	☐	☐	✓	Managed System	☐
		NT_Log_Management	☐	☐	✓	Managed System	☐
		NT_Process_CPU	☐	☐	✓	Managed System	☐
		NT_Process_Memory	☐	☐	✓	Managed System	☐
		OS400_Address_Critical_Message	☐	☐	☐	Managed System	☐
		OS400_Comm_Critical_Message	☐	☐	☐	Managed System	☐
		OS400_High_CPU_Message	☐	☐	☐	Managed System	☐
		OS400_Snads_Critical_Message	☐	☐	☐	Managed System	☐
		UNIX_CPU_Busy	☐	☐	☐	Managed System	☐
		UNIX_Disk_Space_Full	☐	☐	✓	Managed System	☐

Figure 8-16 The workflow editor screen when launched

2. You are asked to name the policy. We call it `ACME_POLICY_oserv_start`.

3. Click the first function we need, **Wait until a situation is true**. It immediately opens a list of the situations in the environment as shown in Figure 8-17. We select the situation that we wrote that checks the status of the oserv.



Undo	Edit Workfl...	Policy name	Distributed	Auto start
		ACME_POLICY_oserv_start	☐	✓

Select a Situation:

- ACME_ITM_Offline_CRITICAL
- ACME_NT_Drwtsn32_CRITICAL
- ACME_TMR_down_CRITICAL
- ACME_UNIX_GWAvail_CRITICAL
- ACME_UNIX_OservAvail_CRITICAL
- HACMP_acquire_service_addr
- HACMP_acquire_takeover_addr
- HACMP_config_too_long

Figure 8-17 Starting our workflow

The function appears in the graphical workspace with a modified label to reflect the situation (Figure 8-18).

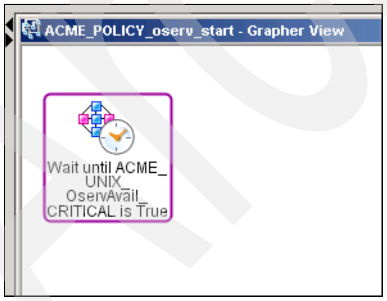
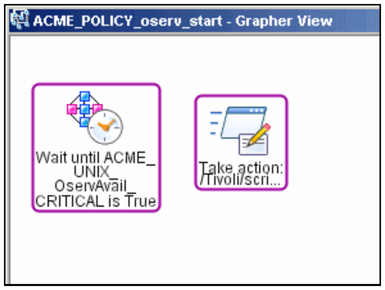

ACME_POLICY_oserv_start - Grapher View
Wait until ACME_UNIX_OservAvail_CRITICAL is True

Figure 8-18 The function created for the oserv situation

4. The next step is to use the function to take action. In this function, specify the command to use to attempt to resolve the issue. We call a shell script (because this is the UNIX oserv) that calls to start the oserv.

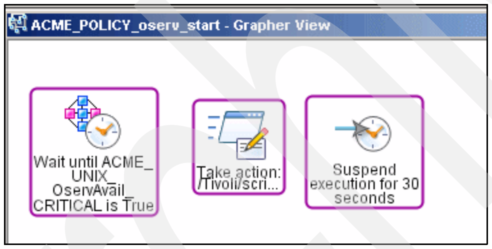
   Figure 8-19 shows our workflow so far.



ACME_POLICY_oserv_start - Grapher View
Wait until ACME_ UNIX_ OservAvail_ CRITICAL is True
Take action: /Tivoli/scri...

Figure 8-19 The workflow so far

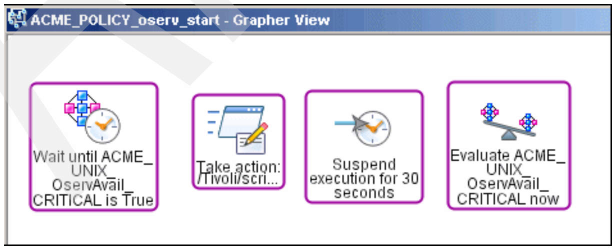
5. Next, we want to give the oserv script a chance to work. We choose to wait for 30 seconds by selecting the function to **Suspend execution**, setting its options for 30 seconds (Figure 8-20).



ACME_POLICY_oserv_start - Grapher View
Wait until ACME_ UNIX_ OservAvail_ CRITICAL is True
Take action: /Tivoli/scri...
Suspend execution for 30 seconds

Figure 8-20 Our workflow at this point

6. Next, we want the situation to reevaluate at this point using **Evaluate a situation now**, again specifying the same situation. (Figure 8-21).



ACME_POLICY_oserv_start - Grapher View
Wait until ACME_ UNIX_ OservAvail_ CRITICAL is True
Take action: /Tivoli/scri...
Suspend execution for 30 seconds
Evaluate ACME_ UNIX_ OservAvail_ CRITICAL now

Figure 8-21 Our workflow nearly complete

7. We can take different actions based on the results. First, we must connect the functions in the workflow using the connector tool. As you connect functions, you are prompted to set the conditions for the flow to pass to the next function. When you reach Evaluate now, your workflow should look like Figure 8-22.

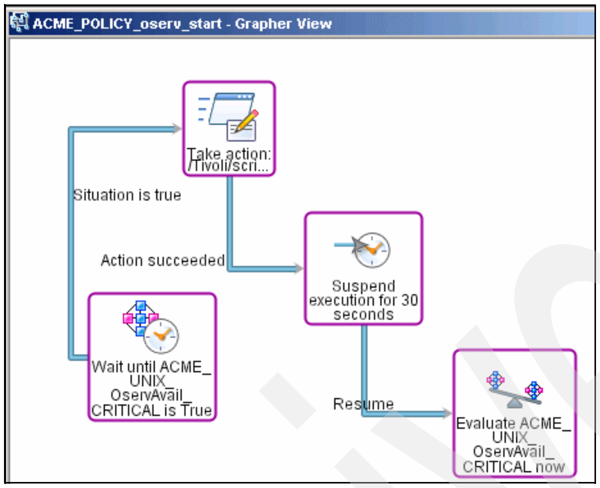

ACME_POLICY_oserv_start - Grapher View
Take action: /Tivoli/scri...
Situation is true
Action succeeded
Suspend execution for 30 seconds
Wait until ACME_UNIX_OservAvail_CRITICAL is True
Resume
Evaluate ACME_UNIX_OservAvail_CRITICAL now

Figure 8-22 Our workflow showing the connections thus far

8. When the situation is evaluated at the function to Evaluate now, we will proceed in one of two ways: when the situation is still true (oserv restart has failed) or when the situation evaluates false (oserv restart succeeded). To this end, we create two new take actions and connect them with the statuses of true and false. Our final policy looks like Figure 8-23.

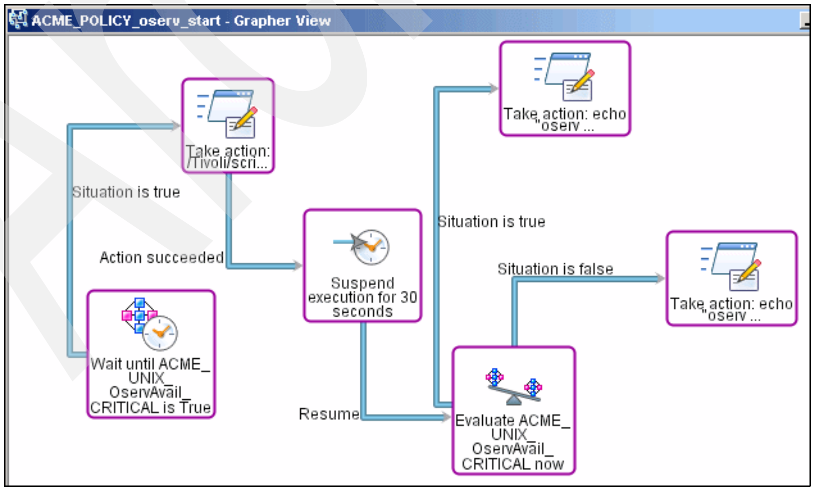

ACME_POLICY_oserv_start - Grapher View
Take action: /Tivoli/scri...
Take action: echo "oserv ...
Situation is true
Action succeeded
Suspend execution for 30 seconds
Situation is true
Situation is false
Take action: echo "oserv ...
Wait until ACME_UNIX_OservAvail_CRITICAL is True
Resume
Evaluate ACME_UNIX_OservAvail_CRITICAL now

Figure 8-23 Our completed workflow policy

Note that there are other options besides a take action that we could have taken such as:

– Do nothing. The situation is false or true and it is now on the console as such.

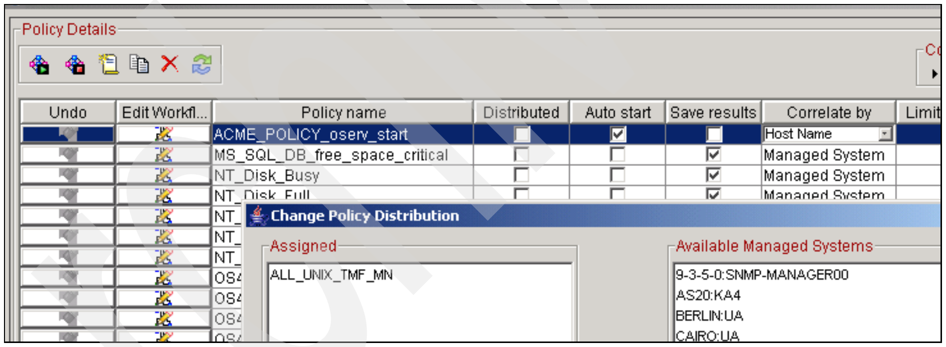
– Choose Emitter for SNMP or TEC. Generate an IBM Tivoli Enterprise Console event or an SNMP trap.

– Choose some other situation to evaluate.

In our example, we merely chose to execute a system command to mail a user with a message. We could have pursued any of the above options.

9. Finally, modify the situation settings to use this policy where we wish to do so. It does not have to be all the systems. We look up at the policy settings and note that the policy has listed to correlate by host name. We are fine with this as all of our policy happens on a single system. It does not have to.

   We select to run it at all managed nodes as this graphic shows when you click the **Distributed** check box. At this point, we go back and edit the situation to remove the action that we took within the situation itself because now we are doing it from the workflow policy (Figure 8-24).



Undo	Edit Workfl...	Policy name	Distributed	Auto start	Save results	Correlate by	Limit
		ACME_POLICY_oserv_start	☐	✓	☐	Host Name	
		MS_SQL_DB_free_space_critical	☐	☐	✓	Managed System	
		NT_Disk_Busy	☐	☐	✓	Managed System	
		NT_Disk_Full	☐	☐	✓	Managed System	

Assigned: ALL_UNIX_TMF_MN

Available Managed Systems:
9-3-5-0:SNMP-MANAGER00
AS20:KA4
BERLIN:UA
CAIRO:UA

Figure 8-24 Selecting the targets for this workflow policy

Next, we set up a managed system list and build a similar monitor for all Tivoli gateways.

8.4.3 Monitoring the Tivoli UNIX gateways

Monitoring the Tivoli gateway is not quite as straightforward as the oserv for a couple of reasons:

- Some gateways might be on systems with other processes that contain the string "gateway," such as the IBM Tivoli Enterprise Console gateway or the Sentry gateway. Our string to compare will have to be more specific.

- If the oserv is down, the gateway will be down automatically. Therefore, we put a note in the expert advice that there might be an associated oserv down event. We could write a rule at IBM Tivoli Enterprise Console to account for this and to close the IBM Tivoli Enterprise Console event associated with the gateway down where there is an associated oserv down event (assuming that we do not do the event synchronization, of course).

We therefore create our missing command to look for /TMF/LCF/gateway instead of *gateway. The following example shows the information about the monitoring we have created.

Example 8-5 The ACME_UNIX_GWAvail_CRITICAL situation

```
[root@milan][/opt/IBM/ITM]-> tacmd viewsit -s ACME_UNIX_GWAvail_CRITICAL
Name                         : ACME_UNIX_GWAvail_CRITICAL
Description                  : checks for running gateway
Type                         : UNIX OS
Formula                      : *IF *MISSING Process.Command *EQ (
'/TMF/LCF/gateway
' )
Sampling Interval            : 0/0:1:0
Run At Start Up              : Yes
Distribution                 : ALL_UNIX_TMF_GW
Text                         : The gateway on
$EVENT:ATTRIBUTE.ISITSTSH.ORIGINNODE$
 is down. Please check for a related event ACME)_UNIX_OservAvail_CRITICAL for
$EVENT:ATTRIBUTE.ISITSTSH.ORIGINNODE$. The Tivoli on-call person has been
paged.
Action Location              : Agent
Action Selection             : System Command
Universal Message Category:
Universal Message Severity:
Universal Message            :
True For Multiple Items      : Action on First item only

[root@milan][/opt/IBM/ITM]-> tmp/ACME_UNIX_GWAvail_CRITICAL_v1_0.xml        <

KUICVS004I: The Situation ACME_UNIX_GWAvail_CRITICAL was exported to
/tmp/ACME_UNIX_GWAvail_CRITICAL_v1_0.xml.

[root@milan][/opt/IBM/ITM]-> cat /tmp/ACME_UNIX_GWAvail_CRITICAL
<TABLE>
<ROW>
<SITNAME>
ACME_UNIX_GWAvail_CRITICAL
</SITNAME>
<TEXT>
<![CDATA[checks for running gateway]]>
</TEXT>
```

```
<AFFINITIES>
00f2000000000000000000000000000##00000w0a7
</AFFINITIES>
<PDT>
<![CDATA[*IF *MISSING Process.Command *EQ ( '/TMF/LCF/gateway' )]]>
</PDT>
<REEV_DAYS>
0
</REEV_DAYS>
<REEV_TIME>
000100
</REEV_TIME>
<AUTOSTART>
*YES
</AUTOSTART>
<ADVISE>
<![CDATA[The gateway on $EVENT:ATTRIBUTE.ISITSTSH.ORIGINNODE$ is down. Please
check for a related event ACME)_UNIX_OservAvail_CRITICAL for
$EVENT:ATTRIBUTE.ISITSTSH.ORIGINNODE$. The Tivoli on-call person has been
paged.]]>
</ADVISE>
<CMD>
<![CDATA[echo "gateway on $EVENT:ATTRIBUTE.ISITSTSH.ORIGINNODE$ down" | mail
tivoncall@sprintpcs.com]]>
</CMD>
<AUTOSOPT>
NNN
</AUTOSOPT>
<DISTRIBUTION>
ALL_UNIX_TMF_GW
</DISTRIBUTION>
<ALERTLIST>
</ALERTLIST>
<HUB>
</HUB>
<QIBSCOPE>
E
</QIBSCOPE>
<SENDMSGQ>
*NONE
</SENDMSGQ>
<DESTNODE>
</DESTNODE>
<LOCFLAG>
</LOCFLAG>
<LSTCCSID>
en_US
</LSTCCSID>
<LSTDATE>
```

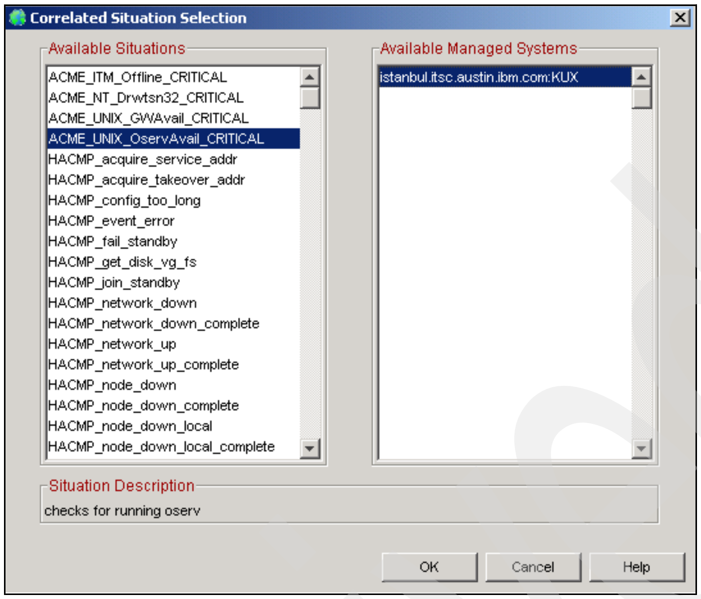
```
1051021143202000
</LSTDATE>
<LSTRELEASE>
V100
</LSTRELEASE>
<LSTUSRPRF>
SYSADMIN
</LSTUSRPRF>
<NOTIFYARGS>
</NOTIFYARGS>
<NOTIFYOPTS>
</NOTIFYOPTS>
<OBJECTLOCK>
</OBJECTLOCK>
<PRNAMES>
</PRNAMES>
<REFLEXOK>
</REFLEXOK>
<SITINFO>
<![CDATA[~]]>
</SITINFO>
<SOURCE>
</SOURCE>
</ROW>
</TABLE>
```

8.4.4 Correlating events across situations

We have just created two situations to monitor the availability of the oservs and the gateway. We create a third event that represents an event when the oserv is down on the TMR. We have to deal with a couple of limitations here:

- At this time, our IBM Tivoli Enterprise Console forwarding does not support correlated events. We will use the action function to call **wpostzmsg** binaries on the TMR and generate an IBM Tivoli Enterprise Console event to the other IBM Tivoli Enterprise Console server (presumably our Hub) as a workaround.
- In order to have a correlated event, there must be more than one situation being evaluated. In this example the oserv and gateway are down on the istanbul system (Figure 8-25 on page 513). It could just as easily be the UNIX oserv down on the TMR and the NT oserv down on a managed node of the TMR.

We are given the specific option of which system the situation gets evaluated on for comparison. This enables us to have a generic "oserv down" situation and yet be able to correlate that it is the oserv and gateway (in our case) down on the same box or oservs across the environment or whatever.


Correlated Situation Selection

Available Situations

ACME_ITM_Offline_CRITICAL
ACME_NT_Drwtsn32_CRITICAL
ACME_UNIX_GWAvail_CRITICAL
ACME_UNIX_OservAvail_CRITICAL
HACMP_acquire_service_addr
HACMP_acquire_takeover_addr
HACMP_config_too_long
HACMP_event_error
HACMP_fail_standby
HACMP_get_disk_vg_fs
HACMP_join_standby
HACMP_network_down
HACMP_network_down_complete
HACMP_network_up
HACMP_network_up_complete
HACMP_node_down
HACMP_node_down_complete
HACMP_node_down_local
HACMP_node_down_local_complete

Available Managed Systems

istanbul.itsc.austin.ibm.com:KUX

Situation Description

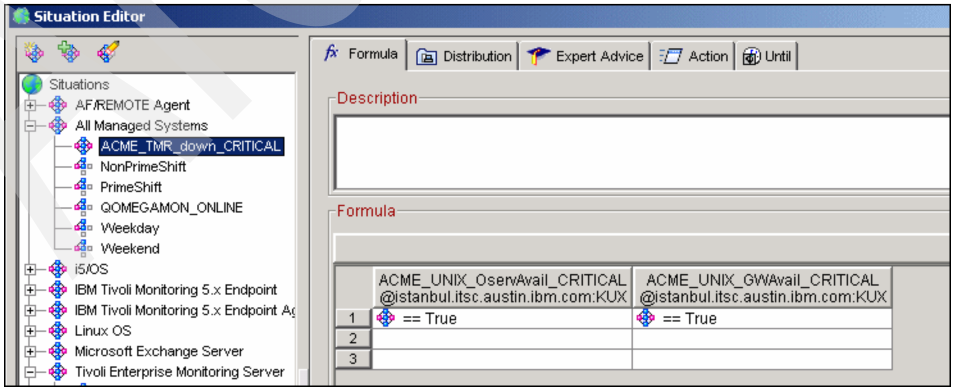
checks for running oserv

Figure 8-25 Selecting the particular system for correlation of an existing situation

Some possible uses of this correlated situation are:

- An event showing that the gateways that service a data center are down
- An event showing that the Hub TEMS has failed to the hot standby (see next section)
- An event showing that the Apache cluster is down (system X and system Y both)

Figure 8-26 shows the creation of a correlated situation.


Situation Editor

Situations
- AF/REMOTE Agent
- All Managed Systems
  - ACME_TMR_down_CRITICAL
  - NonPrimeShift
  - PrimeShift
  - QOMEGAMON_ONLINE
  - Weekday
  - Weekend
- i5/OS
- IBM Tivoli Monitoring 5.x Endpoint
- IBM Tivoli Monitoring 5.x Endpoint Ag
- Linux OS
- Microsoft Exchange Server
- Tivoli Enterprise Monitoring Server

Description

Formula

	ACME_UNIX_OservAvail_CRITICAL @istanbul.itsc.austin.ibm.com:KUX	ACME_UNIX_GWAvail_CRITICAL @istanbul.itsc.austin.ibm.com:KUX
1	== True	== True
2		
3		

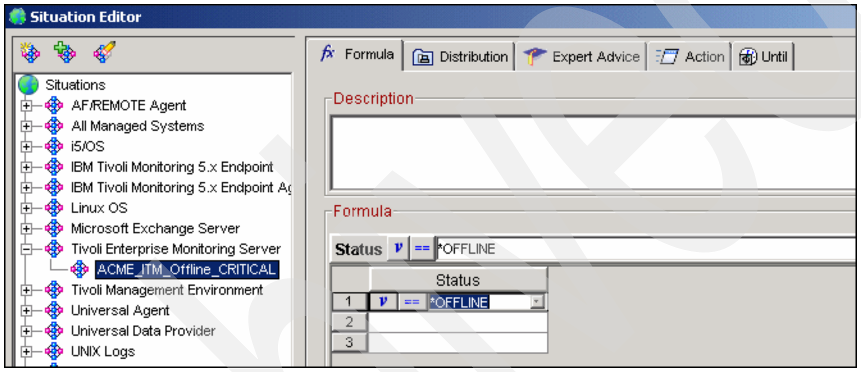
Figure 8-26 The correlated oserv and gateway down on same system

8.4.5 Monitoring the IBM Tivoli Monitoring 6.1 servers

With its separate infrastructure, you also want to monitor the components of the IBM Tivoli Monitoring 6.1 infrastructure: remote TEMS servers and the TEPS server (or servers).

Because there is a heartbeat between the interconnected TEMS systems, we merely need to access that information to obtain the status of all remote TEMS and TEPS from the Hub TEMS.

We create a new situation called `ACME_ITM_Offline_CRITICAL` and select the TEMS as our monitored application. as shown in Figure 8-27.



Situations
- AF/REMOTE Agent
- All Managed Systems
- i5/OS
- IBM Tivoli Monitoring 5.x Endpoint
- IBM Tivoli Monitoring 5.x Endpoint A
- Linux OS
- Microsoft Exchange Server
- Tivoli Enterprise Monitoring Server
  - ACME_ITM_Offline_CRITICAL
- Tivoli Management Environment
- Universal Agent
- Universal Data Provider
- UNIX Logs

Description

Formula

Status == *OFFLINE

	Status
1	== *OFFLINE
2	
3	

Figure 8-27 Building a situation to monitor IBM Tivoli Monitoring 6.1components

This situation will create events to show that IBM Tivoli Monitoring 6.1 components are offline in our TEMS environment.

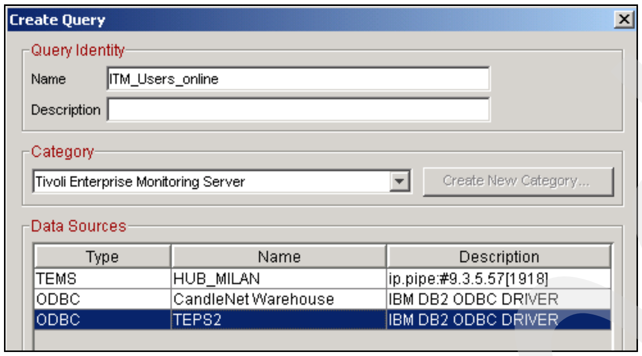
8.4.6 Reporting: monitoring the users logged in to ITM

Another useful bit of information is a list of logged-in IBM Tivoli Monitoring 6.1 users. This information is contained in the TEPS database. In exploring the method for obtaining this information, we can see how to query the information in any database.

First, we identify the database information. In our environment, the TEPS server uses a DB2 database. Using the control center on Windows, we can peruse the database from DB2 perspective to see what kinds of information is available.

We see that the information we need is contained in the KFWLOGIN and KFWUSER tables in the DB2 database.

We create the query that will retrieve the information from the database. To launch the query editor from the Windows TEP, use Ctrl + q. We name the query and select the appropriate data sources (Figure 8-28).



Name: ITM_Users_online
Description:

Category: Tivoli Enterprise Monitoring Server

Data Sources

Type	Name	Description
TEMS	HUB_MILAN	ip.pipe:#9.3.5.57[1918]
ODBC	CandleNet Warehouse	IBM DB2 ODBC DRIVER
ODBC	TEPS2	IBM DB2 ODBC DRIVER

Figure 8-28 Defining the query for which users are using the TEPS

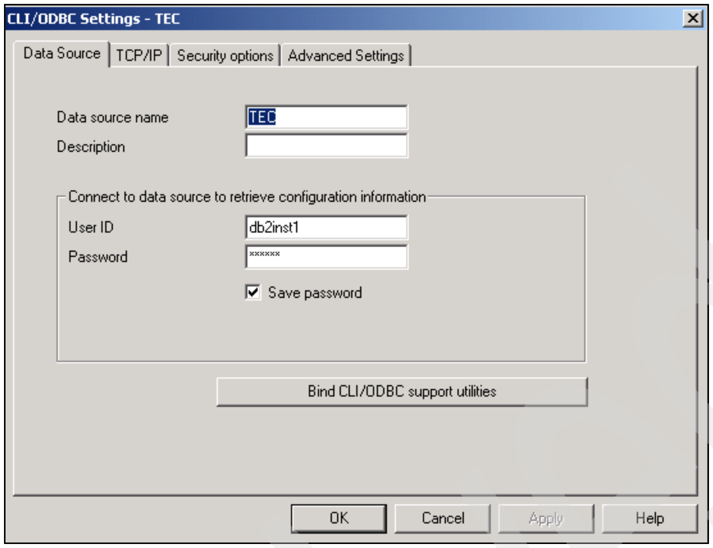
After we define the query, we provide the SQL for the query. In this case, we use **`SELECT * FROM KFWLOGIN`**, which basically returns the entire table. We could just as easily pare it down to an attribute or a few or hide the columns when we use this query to build a report view of this information.

8.4.7 Mining the IBM Tivoli Enterprise Console database

Now we demonstrate how to query a database that is not ITM. The first step is to add the data source to the TEPS. We do this by adding it to the environment file for the TEPS. We use the MTEMS GUI to do this.

Important: So far, your knowledge of IBM Tivoli Monitoring 6.1 might lead you to overlook the other functions and concentrate only on situation. Think about the power of querying external databases for information to display in the workspaces!

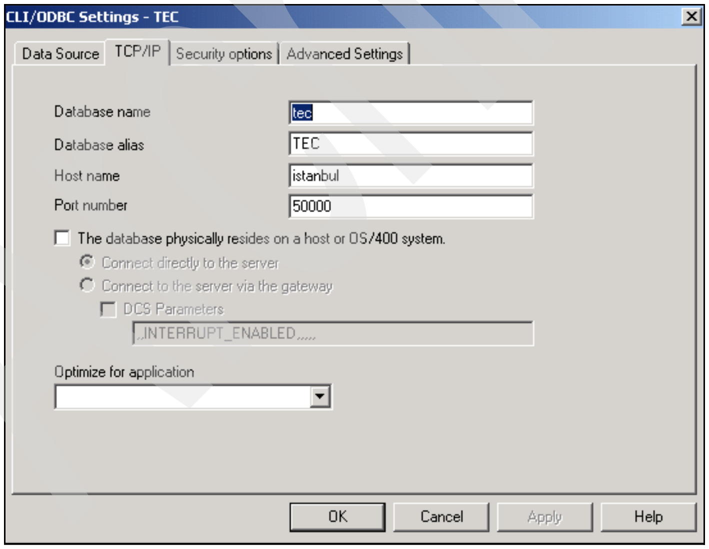
1. Set up an ODBC connection on the Windows TEPS server for the IBM Tivoli Enterprise Console server on istanbul. The following graphics illustrate the correct settings. You will do this through the control panel administrative tools for ODBC data sources management.
2. Name the connection `IBM Tivoli Enterprise Console` (our choice) and configure the information required for the database itself, including user name and password information as shown in Figure 8-29 on page 516.



Data source name: TEC
Description:
Connect to data source to retrieve configuration information
User ID: db2inst1
Password: ******
[✓] Save password

Figure 8-29 The ODBC connection settings for a data source we named TEC

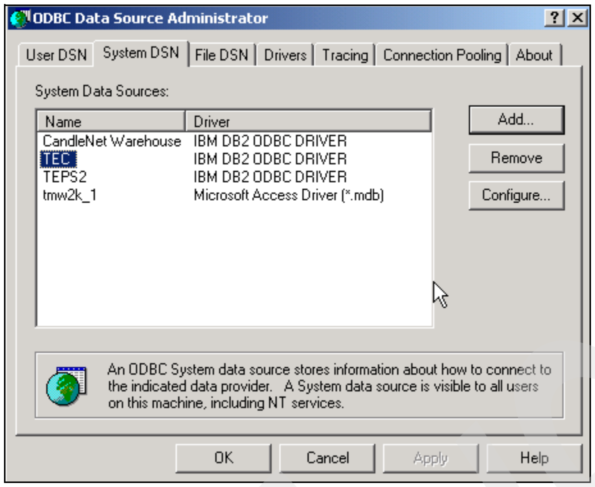
3. Configure the host information for the database server. We must provide the appropriate connection port for the database. DB2 uses 50000 as a default (Figure 8-30).



Database name: tec
Database alias: TEC
Host name: istanbul
Port number: 50000
[ ] The database physically resides on a host or OS/400 system.
(●) Connect directly to the server
( ) Connect to the server via the gateway
[ ] DCS Parameters
,,INTERRUPT_ENABLED,,,,,
Optimize for application:

Figure 8-30 Configuring TCPIP information for istanbul

Figure 8-31 shows the configured data source in the operating system.



System Data Sources:

Name	Driver
CandleNet Warehouse	IBM DB2 ODBC DRIVER
TEC	IBM DB2 ODBC DRIVER
TEPS2	IBM DB2 ODBC DRIVER
tmw2k_1	Microsoft Access Driver (*.mdb)

An ODBC System data source stores information about how to connect to the indicated data provider. A System data source is visible to all users on this machine, including NT services.

Figure 8-31 Our fully configured data source

4. Configure the TEPS server to use the data source from the operating system. The following steps outline the procedure using the Windows MTEMS:

   a. Right-click the TEPS server in the MTEMS GUI.

   b. Select **edit ENV file** from the Advanced menu.

   c. Search for the line that says `DSUSER`.

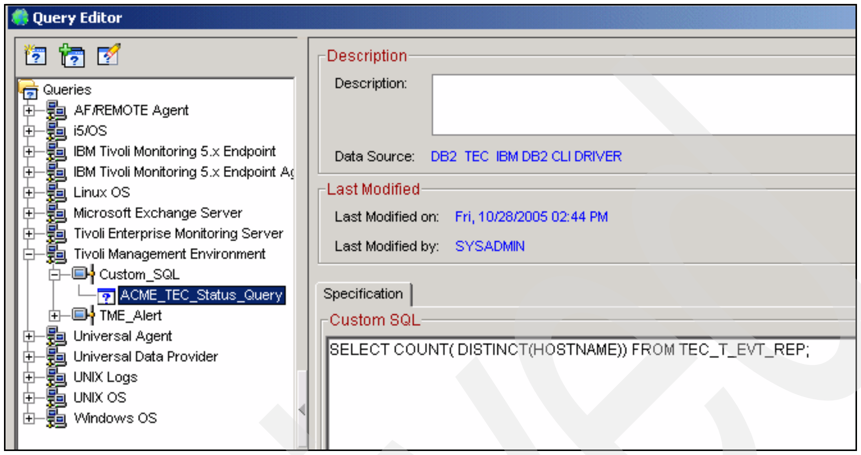
   d. Add the following line:

   ```
   DSUSER1="DSN=TEC;UID=DB2INST1;PWD=itso05"
   ```

   DSN (data source name, equates to the configured data source in the ODBC drivers) matches ODBC, the UID (user ID) is our database user, and PWD (password) is our database password, in this case itso05.

   e. Close the editor, save changes, and restart the TEPS when prompted.

5. Write the query that you want to use. For our query, we will display only the number of unique servers currently having events in the event repository, so we use a fairly simple SQL statement as shown in Figure 8-32.



Description:

Data Source: DB2 TEC IBM DB2 CLI DRIVER

Last Modified on: Fri, 10/28/2005 02:44 PM

Last Modified by: SYSADMIN

```
SELECT COUNT( DISTINCT(HOSTNAME)) FROM TEC_T_EVT_REP;
```

Figure 8-32 Our IBM Tivoli Enterprise Console query SQL statement

8.5 Building a graphical view for a Tivoli administrator

The next task we can undertake is to build a workspace view for a Tivoli infrastructure and IBM Tivoli Monitoring 6.1 administrator. Our goal is to build a business systems view (dashboard) of the Tivoli environment. We could have many different things in this view, but for our purposes we will build something that uses the situations and queries that we have just completed.

Important: If you damage your default workspace too seriously, you will have to reinstall the TEPS server.

The steps to build the dashboard view typically involve:

- 8.5.1, “Building a hierarchy of navigator views” on page 519
- 8.5.2, “Assigning systems to the navigator views” on page 520
- 8.5.3, “Building the workspaces for our views” on page 522

8.5.1 Building a hierarchy of navigator views

1. Open the Edit Navigator View panel to create a new Navigator. (Figure 8-33).

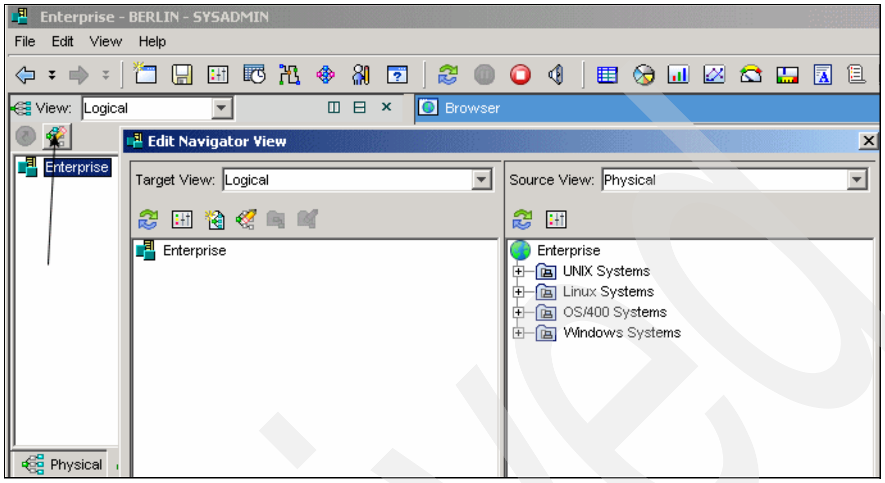


Figure 8-33 Open the Edit Navigator View

Note that the logical navigator is on the left. This is the target navigator and can be modified. The physical navigator is on the right and is the source of all shareable items. It contains the navigator items for all agents that have connected successfully to the Hub TEMS.

2. Select the third button from the left to create a new navigator view (Figure 8-34).

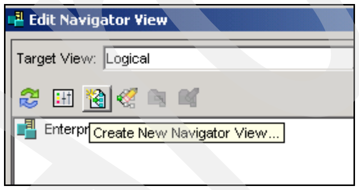
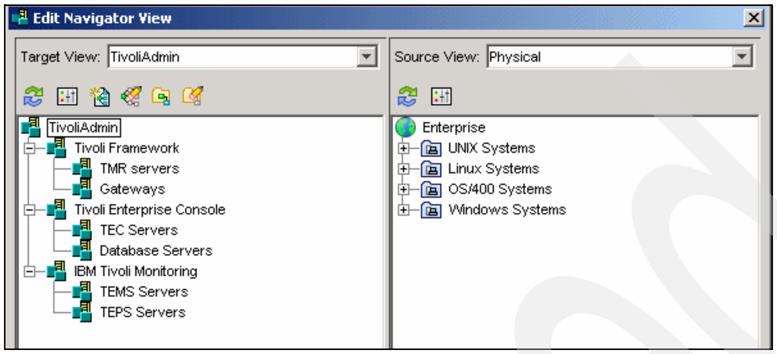


Figure 8-34 Selecting to create a new navigator view

3. We add the navigator name as `TivoliAdmin` and give it a description.
4. Create the structure of the navigator. Because so far we have created situations and queries that cover distinct areas of Tivoli, we will build child items for the TivoliAdmin to cover those areas. When you select the TivoliAdmin navigator item, the Create child button becomes available (Figure 8-35 on page 520).

You can construct a hierarchy of navigator views as we will do as an example. We will not completely create all the workspace views for all of these navigator items in this book.



```
TivoliAdmin
  Tivoli Framework
    TMR servers
    Gateways
  Tivoli Enterprise Console
    TEC Servers
    Database Servers
  IBM Tivoli Monitoring
    TEMS Servers
    TEPS Servers
```

```
Enterprise
  UNIX Systems
  Linux Systems
  OS/400 Systems
  Windows Systems
```

Figure 8-35 The navigator view logical design areas

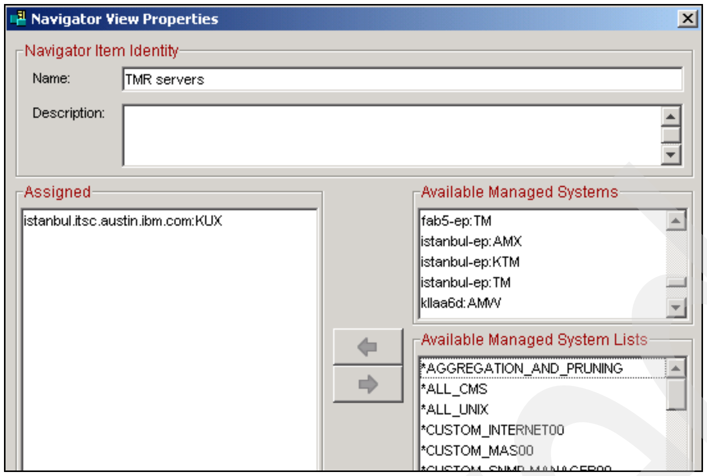
5. When you have completed building this structure, close the navigator editor. The work is automatically saved. Now the navigator is selectable from the navigator pull-down menu.

8.5.2 Assigning systems to the navigator views

The next step is to assign the systems to each of these navigator views. We begin by assigning the TMR server (or servers) in our infrastructure to the TMR servers navigator:

1. Right-click the TMR servers and select **Properties**.
2. Move the TMR servers into the assignment pane (Figure 8-36 on page 521).

   The lowest level of the hierarchy must have systems assigned to it in order to be able to view data from the managed systems and generate situation events. The layers above the bottom level will also have those systems available.


Navigator View Properties

Navigator Item Identity

Name: TMR servers

Description:

Assigned

istanbul.itsc.austin.ibm.com:KUX

Available Managed Systems

fab5-ep:TM
istanbul-ep:AMX
istanbul-ep:KTM
istanbul-ep:TM
kllaa6d:AMW

Available Managed System Lists

*AGGREGATION_AND_PRUNING
*ALL_CMS
*ALL_UNIX
*CUSTOM_INTERNET00
*CUSTOM_MAS00

Figure 8-36 Moving a TMR server to the assigned systems list

3. Repeat this process to add the systems to each navigator area. You have to add the agent from a managed system that will provide the data that you consume in that space.

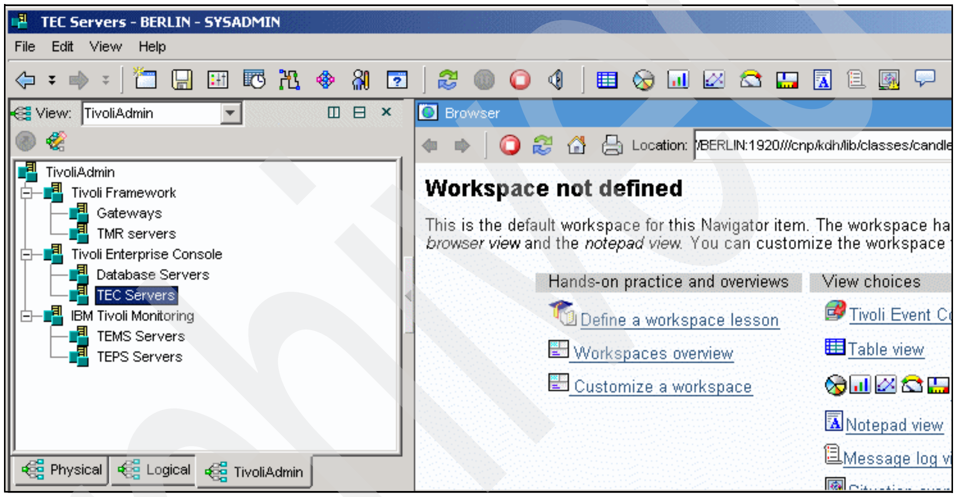
When you have completed this process, you can click on the navigator items and start to build the workspace view for these navigator items.

8.5.3 Building the workspaces for our views

Before starting construction of the workspace views, it is best to plan your workspace on paper. When you are building it in the TEP, it is very difficult to make changes and have the results come out the way you want without starting over from the default.

Each pane of the workspace is built by subdividing the existing space into vertical and horizontal chunks.

1. Click **TEC Servers**. The workspace view changes to an unassigned default state (Figure 8-37).

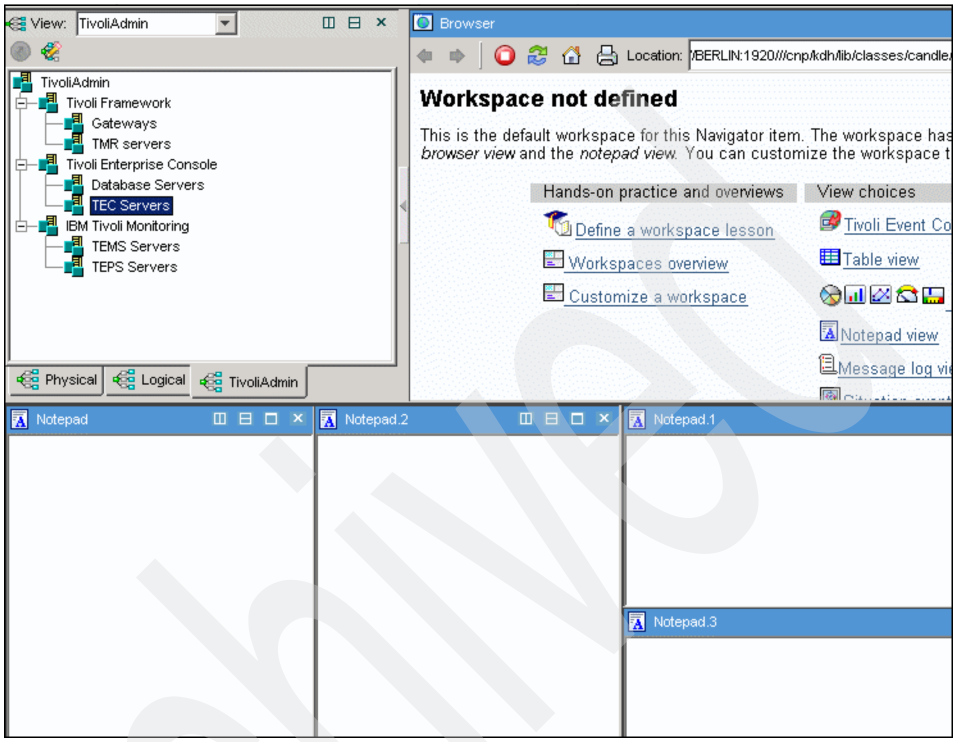


Workspace not defined

This is the default workspace for this Navigator item. The workspace ha
browser view and the *notepad view*. You can customize the workspace

Figure 8-37 Default workspace view for our TEC server's navigator

2. Subdivide the real estate into the desired sections through vertical and horizontal splitting. Our completed design looks like Figure 8-38.



Workspace not defined

This is the default workspace for this Navigator item. The workspace has *browser view* and the *notepad view*. You can customize the workspace t

Figure 8-38 Our workspace divided into planned sections

3. Select components to display in this workspace. The first is the event console viewer. Click the icon for this on the toolbar, then click in the space you want to assign it. That is all that it takes to do the assignment.

4. Use the query that we created that counts how many systems are currently producing IBM Tivoli Enterprise Console events that are in the event repository. We want to display the information in this case as a line chart that shows progress over time.

   To do this, select the plot chart button on the toolbar and click in the pane of this workspace that we have preselected for this. It will ask you if you want to assign the query now. Click **OK**. The resulting dialog box has a button in the center to assign the query (Figure 8-39).

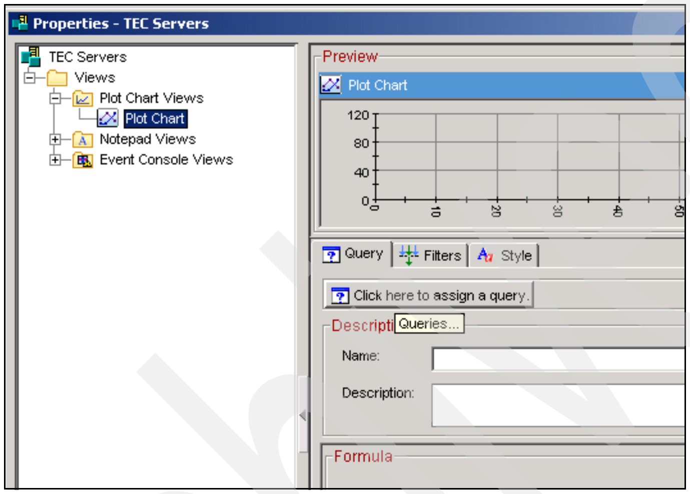


Figure 8-39 Assigning the query to the plot chart

   Use the Style tab to assign text to labels and change appearances of the items on the chart.

Our completed workspace looks like Figure 8-40. Of course, this example is incomplete, but it provides the basis for understanding the construction of workspaces.

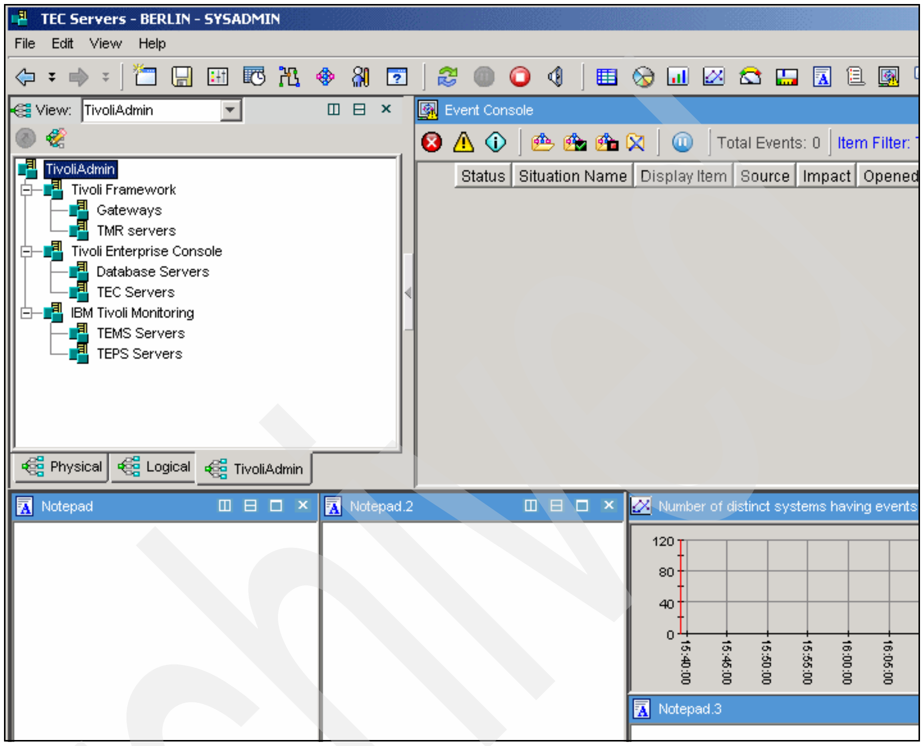


Figure 8-40 Our partially completed workspace for TEC servers

5. Next, we build a graphical view showing the status of our managed nodes. In the navigator, select **Tivoli Framework**. Delete the existing default panels until you are left with the navigator on the left. Resize it to resemble the view in Figure 8-41.

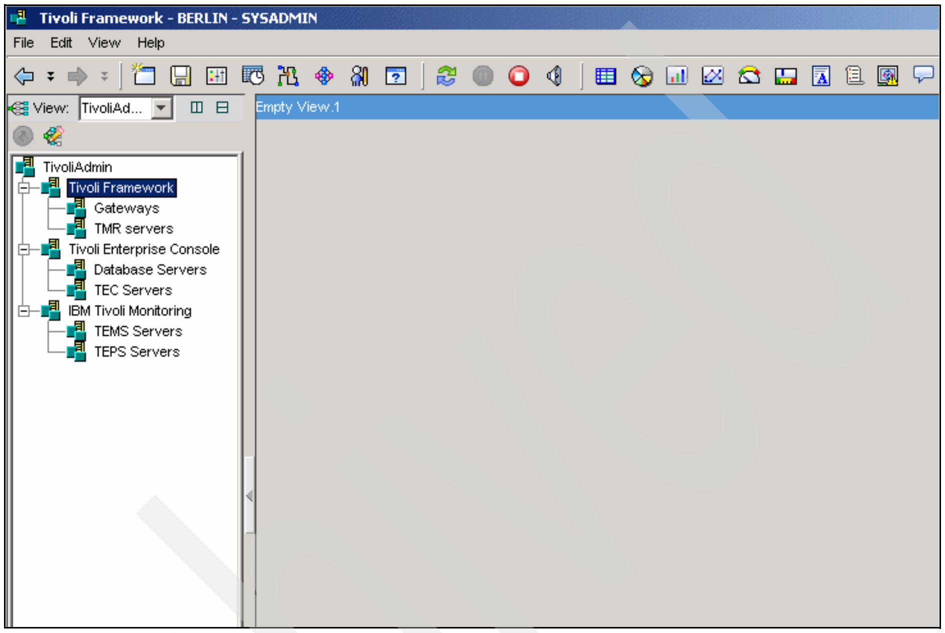


Figure 8-41 The stripped-down default workspace

6. Click the graphic view icon and click in the empty space. A default graphic appears and the child navigator items appear on it. We change the default graphic and the displayed information.
7. First, place the custom graphic in a directory on the TEPS server.

   Create the image you want to use as a background in a graphical editor and save as a .PNG, .JPG, or .GIF file, using a one-word name (underscores are acceptable).

   Copy the image to:

   <install_drive>:\ibm\itm\cnb\classes\candle\fw\resources\backgrounds\user

   It will now be accessible and available to use to replace the default graphic. The online help covers the steps for this particularly well.
8. Drag the managed systems onto the map and use the arrow and delete icons to remove and move around icons to get the view you desire. When you are finished, you will see your custom background as shown in Figure 8-42 on page 527.

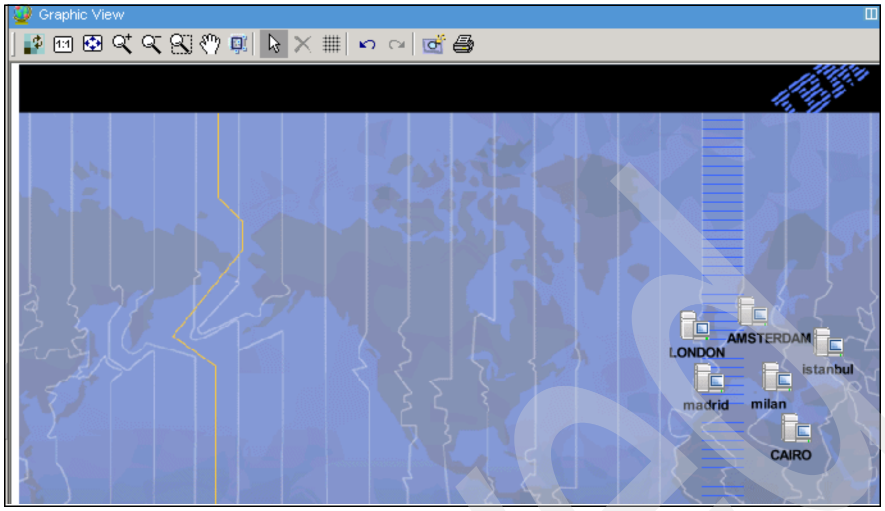
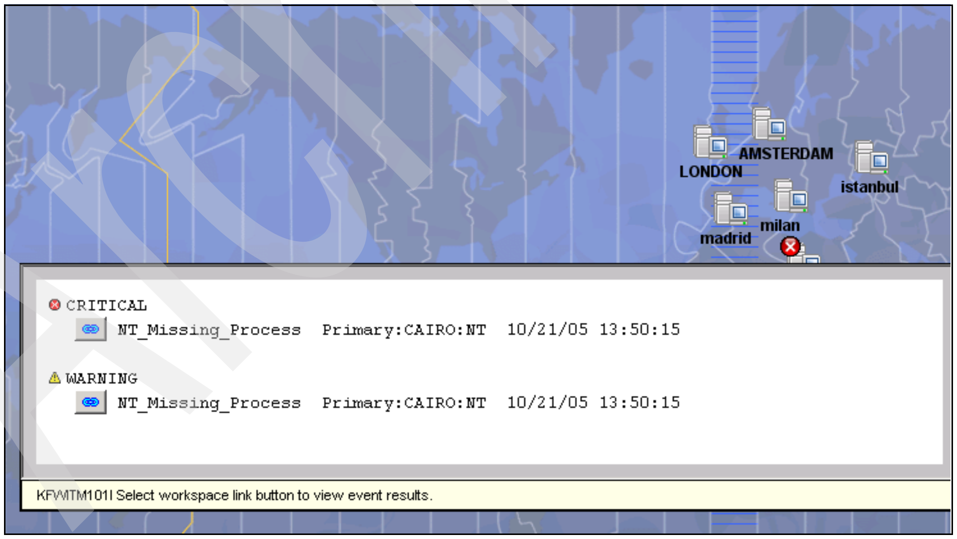


Figure 8-42 Our custom workspace view within the graphic pane

9. When events come on the workspace, you will be able to fly over them (hover) and see the text of the event. Notice how the icons for the resources change just as they would in the navigator to match the severity of the event in Figure 8-43.

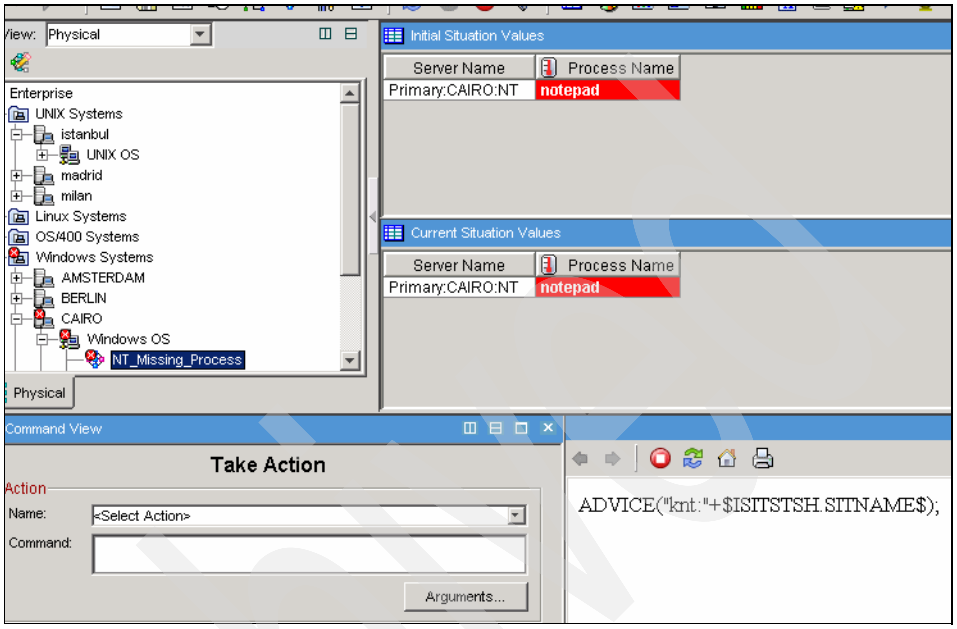


```
CRITICAL
    NT_Missing_Process   Primary:CAIRO:NT   10/21/05 13:50:15

WARNING
    NT_Missing_Process   Primary:CAIRO:NT   10/21/05 13:50:15
```

KFWITM101I Select workspace link button to view event results.

Figure 8-43 Event view in our custom workspace

10. Clicking on these events in the fly-over drills down to the event and launches a linked view to the particular event selected where the expert advice is visible, take actions are possible, etc. This is shown in Figure 8-44.



Server Name	Process Name
Primary:CAIRO:NT	notepad

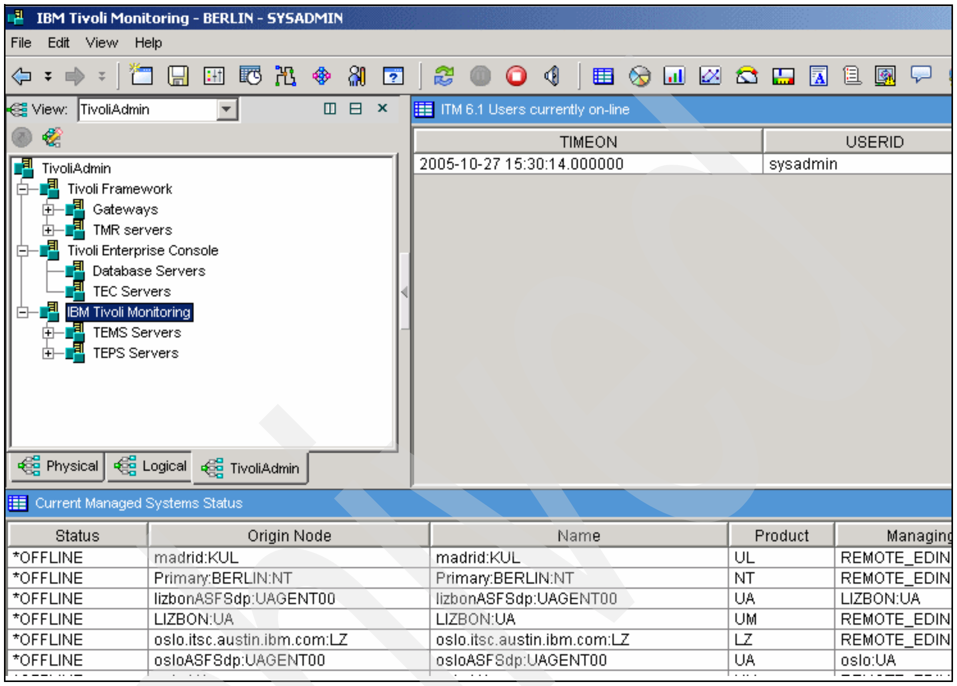
Server Name	Process Name
Primary:CAIRO:NT	notepad

```
ADVICE("knt:"+$ISITSTSH.SITNAME$);
```

Figure 8-44 Drilling down to the view

Note: In Figure 8-44 on page 528, expert advice was broken in this particular build, but normally would show the Web page.

11.Next, we show the ITM View that we have planned. We display in one report pane the query showing those users online, and in the other pane the report showing managed system status. The final result looks like Figure 8-45.



ITM 6.1 Users currently on-line

TIMEON	USERID
2005-10-27 15:30:14.000000	sysadmin

Current Managed Systems Status

Status	Origin Node	Name	Product	Managing
*OFFLINE	madrid:KUL	madrid:KUL	UL	REMOTE_EDIN
*OFFLINE	Primary:BERLIN:NT	Primary:BERLIN:NT	NT	REMOTE_EDIN
*OFFLINE	lizbonASFSdp:UAGENT00	lizbonASFSdp:UAGENT00	UA	LIZBON:UA
*OFFLINE	LIZBON:UA	LIZBON:UA	UM	REMOTE_EDIN
*OFFLINE	oslo.itsc.austin.ibm.com:LZ	oslo.itsc.austin.ibm.com:LZ	LZ	REMOTE_EDIN
*OFFLINE	osloASFSdp:UAGENT00	osloASFSdp:UAGENT00	UA	oslo:UA

Figure 8-45 Our ITM information panel that we created

In Chapter 13, “Administration and maintenance” on page 697, we cover how to publish workspaces to users.

9

IBM Tivoli Universal Agent

The IBM Tivoli Universal Agent (Universal Agent or UA) is a generic agent used in conjunction with other Tivoli Enterprise Monitoring Agents to collect data and monitoring systems and applications in your network. In this chapter, we cover the following IBM Tivoli Universal Agent topics:

- What is IBM Tivoli Universal Agent?
- IBM Tivoli Universal Agent architecture
- Universal Agent deployment scenarios

9.1 What is IBM Tivoli Universal Agent?

The IBM Tivoli Universal Agent is a generic agent used to collect data and monitoring systems and applications in your network. In turn, this data can be used and visualized in the Tivoli Enterprise Portal. You can use all standard TEP data viewing options with the Universal Agent.

It is very important to understand the difference between standard Tivoli Enterprise Monitoring Agents and IBM Tivoli Universal Agent, because these two types of agents complement each other to provide a robust and completely flexible monitoring solution. Tivoli Enterprise Monitoring Agents use a static set of hard-coded attributes (in other words, predefined data), so they cannot be enhanced by the field personnel to “see” more than they are developed for. The Universal Agent is a full-featured IRA (intelligent remote agent) with dynamic application capabilities. Using the Universal Agent, you can dynamically create custom attributes and catalogs. It adds to monitoring solutions to make them complete and flexible for all platforms.

Most applications and systems have additional information that can be discovered by looking through the log files or using custom programs to query them. By combining this information with information collected by IBM Tivoli Universal Agents, we can generate better alerts to tell more about the health of an installation. The data collected by the Universal Agents can be used for specific monitoring actions through the Tivoli Enterprise Portal *situations* and *policies*.

Benefits of using the Universal Agent include:

- Monitors only the data attributes that interest you (configured through *metafile* applications).
- Enables you to respond quickly to changing monitoring and management scenarios. For example, changes in the metafile can be easily made to support new features in an application.
- Monitors data not supported by other Tivoli Enterprise Monitoring Agents.
- Integrates data from virtually any operating system and any source.
- Gives you control of attributes and surfacing of data.
- Provides a means of agentless monitoring.

9.2 IBM Tivoli Universal Agent architecture

Figure 9-1 shows the high-level architecture and data flow for the Universal Agent.

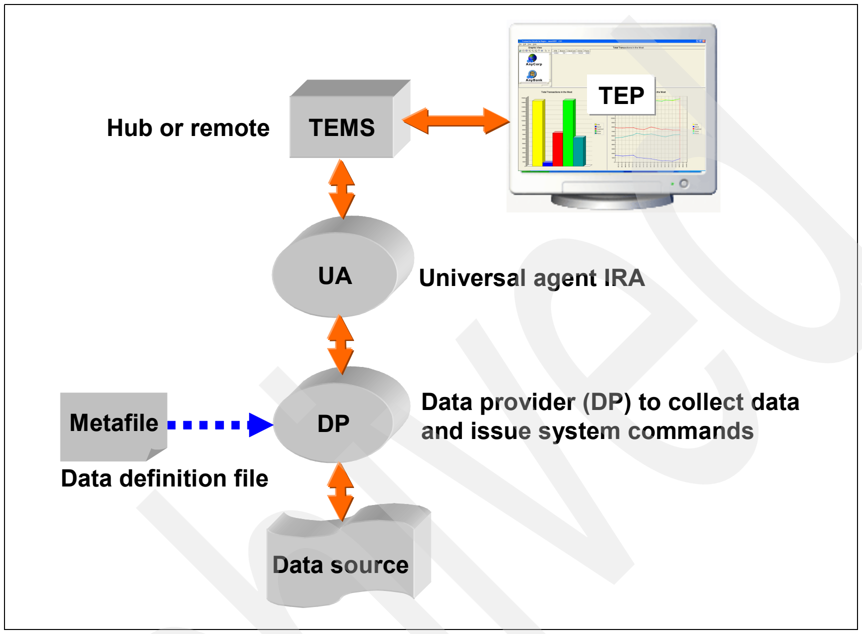


Figure 9-1 Universal Agent high-level architecture and data flow

The data source for the Universal Agent is something that the installation provides. It could be a log file, a script, an ODBC data source, or a program that the site creates or customizes to feed data to the Universal Agent.

Metafiles map out data that is coming into a Universal Agent. They are used to define the data structure to be monitored.

Data providers serve as the data interfaces for the Universal Agent; in other words, they are the “ears” of the Universal Agent.

Data collected by the Universal Agents can be monitored (and used in situations) through the Tivoli Enterprise Portal, just like the data collected by Tivoli Enterprise Monitoring Agents.

Data providers, Universal Agent, and the IBM Tivoli Monitoring Agents can all reside on the same machine or they can be separated as the situation requires. Although it is useful from a conceptual standpoint to view data providers as autonomous entities, they normally run as threads inside the main UA process.

It is possible to run more than one Universal Agent on a host, but it is generally not necessary, because one Universal Agent can monitor data from multiple SNMP agents, ODBC data sources, API clients, scripts, and socket clients.

9.2.1 Data provider: Informing IBM Tivoli Universal Agent how to collect and monitor

Data is collected from the monitoring environment and passed to IBM Tivoli Universal Agent through structures named *data providers*. Data providers work as IBM Tivoli Universal Agent threads, using applications named metafile to define the structures to be monitored. Data providers can be analyzed as IBM Tivoli Universal Agent autonomous entities. They are used to define how data is collected from systems and hosts.

Data providers enable the following activities:

- Validate and load metafile applications.
- Collect data from different sources, such as logs, URLs, and SNMP agents.
- Pass collected data and information about metafile definitions to IBM Tivoli Universal Agent

We can choose from nine different data provider categories depending on our monitoring requirements. These are: API Server, API-Socket-File-Script (ASFS), File, HTTP, ODBC, Post, Script, SNMP, and Socket. Table 9-1 lists data providers available with IBM Tivoli Universal Agent.

Table 9-1 IBM Tivoli Universal Agent data providers

Type	Description
API Server	Enables you to collect data from resources on remote machines where the IBM Tivoli Universal Agent API client software is supported.
API, Socket, File, Script (ASFS)	Consolidates four types of data providers into one package, which is started as a single thread to save resource usage. This is the default data provider when you install the IBM Tivoli Universal Agent.
File	Monitors sequential files, such as system or message logs. Provides the most direct, simplest method of collecting data.

Type	Description
HTTP	Enables monitoring of Internet URLs for availability and response time. You can specify URLs to monitor in a startup configuration file or within Tivoli Enterprise Portal situations.
ODBC	Enables data collection from ODBC-compliant databases using SQL Select statements and stored procedures. Only available on Windows.
Post	TCP/IP socket application with predefined data. Enables you to send ad hoc notifications such as messages, alerts, and status.
Script	Enables data collection from any script or program that sends results to standard output.
SNMP	Provides the functionality of an SNMP manager, including network discovery, trap monitoring, and MIB data collection.
Socket	Listens on a TCP/IP socket for data sent using program-to-program communication. Enables you to collect data from remote devices or machines for which no IBM Tivoli Universal Agent API support is available.

The right choice of a data provider depends on the type of data that we want to monitor and the source of the data. For example, if the operational system is z/OS, it might not be possible to use an API data provider. In this case, a better choice would be to use a Socket data provider. Table 9-2 lists data source and related data providers.

Table 9-2 Data source and preferred data providers

Data source	Preferred data providers
Log files	File
Ad hoc notifications such as messages, alerts, and status information	Post
Application internals (supported API client operating system)	API Server
Application internals (non-supported API client operating system) using TCP/IP	Socket

Data source	Preferred data providers
Any combination of the following: Log files: ► Application internals (supported API client operating system) ► Application internals (non-supported API client operating system) ► Stdout messages produced by a script or program	API, Socket, File, Script (ASFS)
Internet or intranet URLs	HTTP
Relational Databases	ODBC
SNMP MIB data	SNMP
Stdout messages produced by a script or program	Script

Note: ASFS is the default data provider setting in the Universal Agent. It consolidates four types of data providers (API, Socket, File, and Script) into one package, which is started as a single thread to save resource usages. This is the default data provider when you install the IBM Tivoli Universal Agent.

The Universal Agent has the ability to run several instances of the same data provider on the same monitored host. The reasons for this could be:

- Run test and production versions of the Universal Agent on the same host.
- Balance the data load of an IBM Tivoli Universal Agent that is overloaded.
- Connect to several IBM Tivoli Universal Agents on different Tivoli Enterprise Monitoring Servers.

Universal Agent and its data providers are configured to communicate over a variety of ports. Every port is changeable in the KUMENV file specifying the correct variable. Table 9-3 lists the typical Universal Agent ports.

Table 9-3 Typical ports used by the IBM Tivoli Universal Agent

Port	Description	Variable
161	Standard SNMP port (used when running SNMP Universal Agent)	-
1919	Data Clearing House port	KUMA_DCH_PORT
7500	Socket data provider listening port	KUMP_DP_PORT

Port	Description	Variable
7575	Post data provider listening port	KUMP_POST_DP_PORT
7600	API data provider listening port	KUMP_API_DPAPI_PORT
7700-7710	Console ports (one for each DP activated at startup)	-
162	SNMP trap monitoring listening port	KUMP_SNMP_TRAP_PORT

By default, console commands target the primary Universal Agent using console port 7700. We can change this port to access a secondary Universal Agent using the KUMP_DPCONSOLE_PORT variable to specify the alternate port number.

9.2.2 Metafiles: Informing Universal Agent what to collect and monitor

With applications called *metafiles,* we define the data structure to be monitored. In other words, metadata is a data map that specifies data characteristics based on application knowledge and monitoring requirements. It splits the input data into fields called attributes that can then be viewed or referenced in situations.

Note: You can have many metafiles: one for each separate data source and type.

Using metafiles, the Universal Agent knows what to monitor on the systems and hosts. After a metafile is defined, it is imported into the Universal Agent and used by data providers that relay collected data to the Universal Agent. This data is finally used by Tivoli Enterprise Monitoring Server (TEMS) similar to data collected by specific IBM Tivoli Monitoring Agents.

Make a metafile application consisting basically of the following items:

- Name of the application
- Name of each application attribute group
- Source or data sources in each group
- Names and characteristics of each attribute item
- Optional application help text, attribute group, and attributes

A metafile has the control statements (if present) displayed in Table 9-4 on page 538.

Table 9-4 Metafile control statement

Control statement	Description
SNMP	For SNMP Data Providers only, introduces the data definition for IBM Tivoli Monitoring–provided SNMP MIB applications. SNMP TEXT introduces the data definition for user-defined SNMP applications.
APPL	Specifies the name that IBM Tivoli Monitoring uses for the application.
NAME	Defines the name of an attribute group, the type of data being collected, and the period for which the data is valid.
INTERNAL	Provides for data redirection between attribute groups as a way to perform additional processing.
SOURCE	Defines the location of the data you are collecting.
RECORDSET	For File Data Providers only, defines the set of records from which the data provider extracts data.
CONFIRM	For Socket Data Providers only, specifies the requirements for data acknowledgment.
SQL	For ODBC Data Providers only, defines the Select statement or stored procedure to use for collecting relational data.
SUMMARY	Defines the requirements for gathering the frequency of data input during monitoring.
ATTRIBUTES	Introduces the attribute definitions and specifies the attribute delimiters in the data string. Below the ATTRIBUTES control statement, list the individual attribute definition statements.

Example 9-1 shows a sample of a metafile that maps out log files. Each log file is identified as a separate managed system. “TAIL” tell the Universal Agent that we are going to read records from the end of the file as they are written.

Example 9-1 A metafile example

```
//appl MVS
//name SYSTEM E
//source file D:\UA_LOGS\PRA1.log TAIL ManagedSystemName=PRA1
//source file D:\UA_LOGS\PRB1.log TAIL ManagedSystemName=PRB1
//source file D:\UA_LOGS\PRC1.log TAIL ManagedSystemName=PRC1
//source file D:\UA_LOGS\PRE1.log TAIL ManagedSystemName=PRE1
//source file D:\UA_LOGS\PRF1.log TAIL ManagedSystemName=PRF1
//source file D:\UA_LOGS\PRG1.log TAIL ManagedSystemName=PRG1
//source file D:\UA_LOGS\PRX1.log TAIL ManagedSystemName=PRX1
//source file D:\UA_LOGS\PRZ1.log TAIL ManagedSystemName=PRZ1
```

```
//attributes ';'
System D 10
Application D 10
Date D 10
Time D 10
Message D 256
Threshold D 10
AutoAction D 20
```

Another point to take into account is the versioning of metafiles. Versioning enables us to identify the level of metafiles and run different versions of metafiles in different systems (for example, to monitor data for a new application version that the old one does not have).

Metafile has both: version and modification number. When it is imported for the first time in the IBM Tivoli Universal Agent, it is assigned a version number 0 and a modification number 0. When changes are made in the metafile and it is refreshed on the IBM Tivoli Universal Agent, the version or modification number is incremented by one, depending the type of the modification.

Changes that do not affect the number of version or modification of the metafile include:

- TTL value
- A change to the SOURCE statement
- Data type from P, S, or K to any of P, S, or K
- Delimiter specified in the ATTRIBUTE statement
- A change to the RECORDSET statement
- A change to the CONFIRM statement
- A change to an attribute FILTER parameters
- A change to the SQL statement

The following changes affect the modification number (minor changes):

- Adding a new attribute to the end of the attribute list for an attribute group
- Adding a new attribute group at the end of the metafile
- Adding, removing, or changing help text
- Atomizing an existing attribute
- Adding, removing, or changing Scale or Precision values
- Adding, removing, or changing Caption values
- Adding, removing, or changing Warehouse or Aggregation parameters
- Adding, removing, or changing HistoricalTimestamp or PrimaryKey options

The following changes increment the version number (major changes):

- Renaming or deleting an existing attribute

- Changing the type of an attribute

- Changing the length of an attribute
- Changing the name of an attribute group
- Adding a new attribute anywhere other than the end of a list of existing attributes
- Changing the order of attributes
- Changing a data type from E to P, S, or K
- Changing a data type from P, S, or K to E
- Adding a new attribute group anywhere other than the end of a metafile

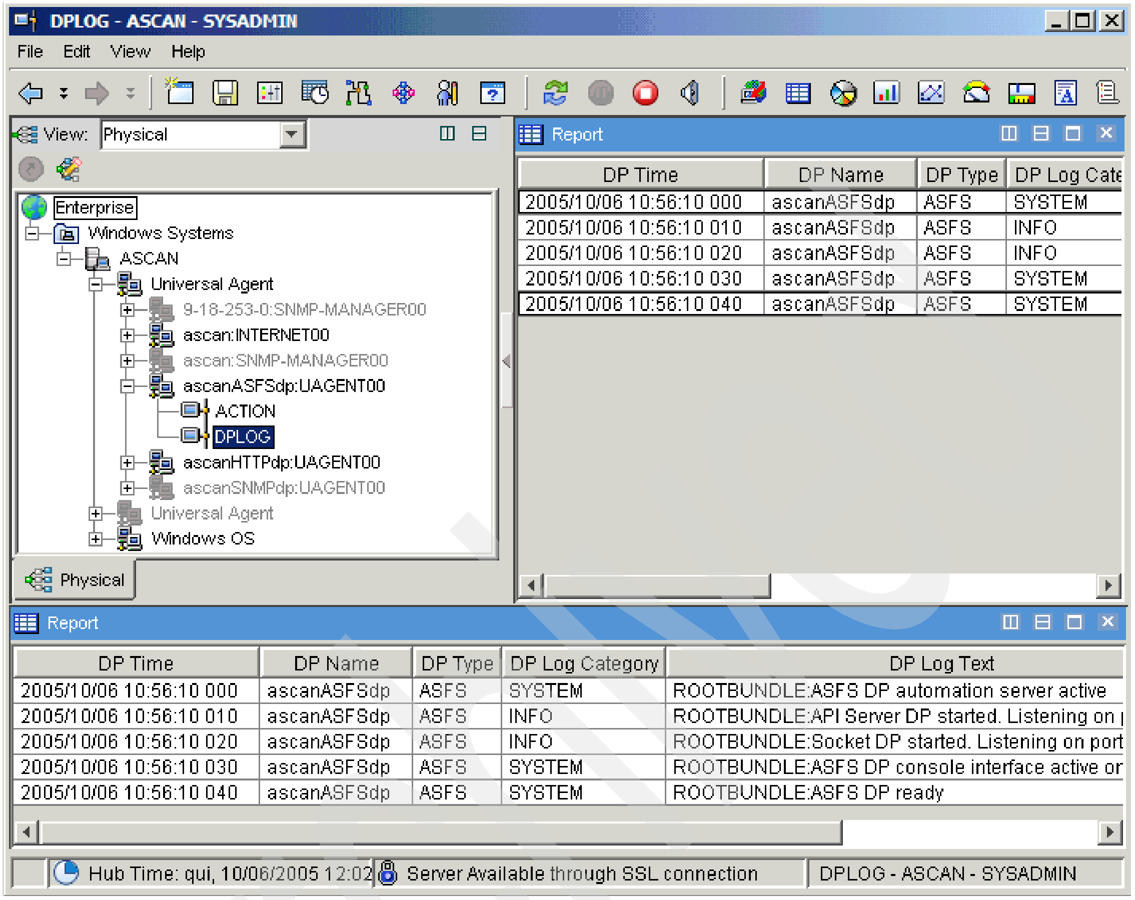
Creating metafiles will be much clearer when you walk through the scenarios in 9.3, “Universal Agent deployment scenarios” on page 543.

9.2.3 Manipulating data with Tivoli Enterprise Portal

The data collected and monitored by the Universal Agent is used in the same way as the data collected by IBM Tivoli Monitoring Agents in the Tivoli Enterprise Portal.

Tivoli Enterprise Portal objects are named managed systems and the name of each managed system identifies the collected data source, application that has been monitored, and metafile version. The Tivoli Enterprise Portal can configure workspaces to visualize collected data by the Universal Agent. Each attribute group defined in a metafile has your own workspace. It can also be customized to show only wanted data.

The attribute groups DPLOG and ACTION from each data provider are used to the Universal Agent self monitoring, more specifically data providers. The DPLOG attribute shows the status from a data provider, and the ACTION attribute gives information about the execution of a situation and policies.



DPLOG - ASCAN - SYSADMIN

Report

DP Time	DP Name	DP Type	DP Log Cate
2005/10/06 10:56:10 000	ascanASFSdp	ASFS	SYSTEM
2005/10/06 10:56:10 010	ascanASFSdp	ASFS	INFO
2005/10/06 10:56:10 020	ascanASFSdp	ASFS	INFO
2005/10/06 10:56:10 030	ascanASFSdp	ASFS	SYSTEM
2005/10/06 10:56:10 040	ascanASFSdp	ASFS	SYSTEM

Report

DP Time	DP Name	DP Type	DP Log Category	DP Log Text
2005/10/06 10:56:10 000	ascanASFSdp	ASFS	SYSTEM	ROOTBUNDLE:ASFS DP automation server active
2005/10/06 10:56:10 010	ascanASFSdp	ASFS	INFO	ROOTBUNDLE:API Server DP started. Listening on
2005/10/06 10:56:10 020	ascanASFSdp	ASFS	INFO	ROOTBUNDLE:Socket DP started. Listening on port
2005/10/06 10:56:10 030	ascanASFSdp	ASFS	SYSTEM	ROOTBUNDLE:ASFS DP console interface active or
2005/10/06 10:56:10 040	ascanASFSdp	ASFS	SYSTEM	ROOTBUNDLE:ASFS DP ready

Hub Time: qui, 10/06/2005 12:02 | Server Available through SSL connection | DPLOG - ASCAN - SYSADMIN

Figure 9-2 Attribute group DPLOG in Tivoli Enterprise Portal

An IBM Tivoli Monitoring situation is a logical expression used with one or more monitoring conditions to monitor a system. The situation editor permits the use of a situation in the following ways:

- Create a situation.
- Save a situation.
- Show a situation.
- Edit a situation.
- Start, stop, and delete a situation
- Verify a situation event at the event workspace.

In the Tivoli Enterprise Portal we visualize the data collected and stored by the Universal Agent with the Historical Data Collection functionality.

The Historical Data Collection enables you to:

- Specify the attribute group or groups for which data is collected.
- Specify the interval at which data is collected.
- Specify the interval at which data is warehoused (if warehouse is being used).
- Determine the source where collected data is stored, in the agent or Tivoli Enterprise Monitoring Server

Basically, Tivoli Enterprise Portal enables you to:

- Visualize stored data or data in real time.
- Define situations with defined thresholds for potential availability or performance problems.
- Define automatic responses for events and levels of alerts from monitored systems.
- Self-monitoring of data providers.

9.2.4 When is using the Universal Agent is a good choice?

The Universal Agent is a good choice, for example: when systems and applications cannot be monitored by existing monitoring solutions, when we want control over monitored data, when the solution needs automation, and when the application to be monitored frequently changes (for example, new applications or operational systems releases).

Tips to consider before the Universal Agent deployment:

- Choose the right data provider for your application monitoring. For example, you can use the ODBC data provider if you want to monitor a relational database, file data provider if you want to monitor log files from an application, and so on.
- Prepare the data source.
- Define an application (metafile) to be used by the data provider that satisfies the monitoring requirements.
- Create situations and policies using IBM Tivoli Monitoring Agent application attributes (metafile attributes), that fires by some systems monitoring conditions.

Some real-world Universal Agent usage examples are:

- Monitoring MQ Series Client Channels
- Monitoring DEC OpenVMS
- Integrating Cabletron Spectrum
- Monitoring remote RF devices

- Monitoring POS devices
- Monitoring proprietary applications

9.3 Universal Agent deployment scenarios

Deploying a Universal Agent solution consists of the following steps:

1. Collect all required information about the solution.
2. Select the data provider and start the UA with selected DP's.
3. Create the metafiles describing the UA application.
4. Load the metafile and send the data.
5. Use standard IBM Tivoli Monitoring features to finalize the solution.

The first step is especially important. Here are some questions that might help you determine to collect all required information about the solution:

- Who needs the information?
- What information is needed?
- Where is the data located?
- When and how often is the data collected?
- Why is it required? Does it make good business sense to collect it?
- How? What methodology will be used to collect the data?
- What is the data used for after it is integrated?

After getting all this information, determine the correct data provider type to use. This decision will be based in the information that you collected in the first step.

In this section we walk you through several scenarios to configure the Universal Agent monitoring using the following data providers:

- HTTP data provider
- File data provider
- ODBC data provider

9.3.1 HTTP data provider deployment scenario

The IBM Tivoli Monitoring HTTP data provider provides the ability to monitor URLs, in some conditions as availability and response time. Different from other data providers, HTTP data provider does not use a definition file (metafile).

Starting HTTP data provider

We start HTTP data provider the same way we start other data providers. For example, with the KUMA_STARTUP_DP parameter in the KUMENV file:

```
KUMA_STARTUP_DP=HTTP
```

To set this value (and start the HTTP data provider), we use the graphical user interface.

Note: It is also possible to start the HTTP data provider (or other data providers) as a separate process. This is useful when data collection has to occur:

- Outside a firewall
- On a special machine with limited resources
- To monitor a file on a remote system

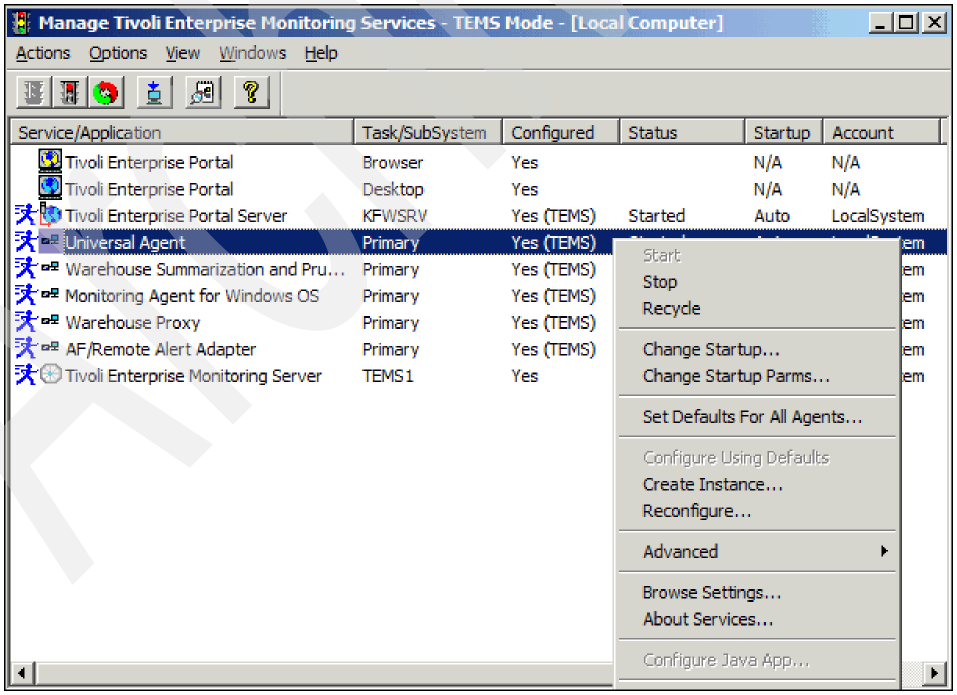
To start HTTP data provider as separate process, invoke this program:

```
KUMPHTTP
```

To start the HTTP data provider from the GUI:

1. In the Manage Tivoli Enterprise Monitoring Services window, right-click **Universal Agent** and select **Reconfigure** as shown in Figure 9-3.

Note: This section is based partly on documentation owned by Stefan Muller.

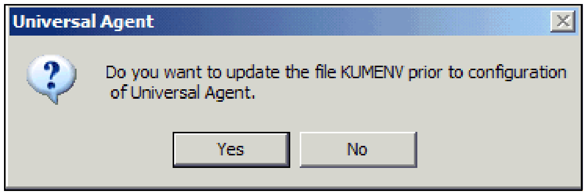


Manage Tivoli Enterprise Monitoring Services - TEMS Mode - [Local Computer]

Service/Application	Task/SubSystem	Configured	Status	Startup	Account
Tivoli Enterprise Portal	Browser	Yes		N/A	N/A
Tivoli Enterprise Portal	Desktop	Yes		N/A	N/A
Tivoli Enterprise Portal Server	KFWSRV	Yes (TEMS)	Started	Auto	LocalSystem
Universal Agent	Primary	Yes (TEMS)			em
Warehouse Summarization and Pru...	Primary	Yes (TEMS)			em
Monitoring Agent for Windows OS	Primary	Yes (TEMS)			em
Warehouse Proxy	Primary	Yes (TEMS)			em
AF/Remote Alert Adapter	Primary	Yes (TEMS)			em
Tivoli Enterprise Monitoring Server	TEMS1	Yes			em

Figure 9-3 Changing data providers startup parameters

2. In the next two windows, click **OK**.
3. When prompted to update the KUMENV file to configure Universal Agent, click **Yes (**Figure 9-4**)**.



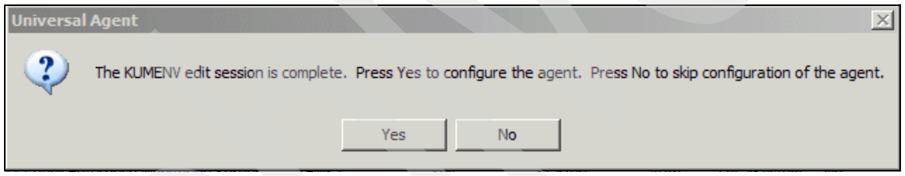
Do you want to update the file KUMENV prior to configuration of Universal Agent.

Figure 9-4 Click Yes to update KUMENV file

4. Search for the line KUMA_STARTUP_DP. Type `HTTP` at the end of the line, save the file, and close it. For example:

```
KUMA_STARTUP_DP=asfs,HTTP
```

5. In the next window, click **Yes** to configure Universal Agent. It will be stopped but not configured yet (Figure 9-5).



The KUMENV edit session is complete. Press Yes to configure the agent. Press No to skip configuration of the agent.

Figure 9-5 Click Yes to configure the Universal Agent

Configuring URL monitoring

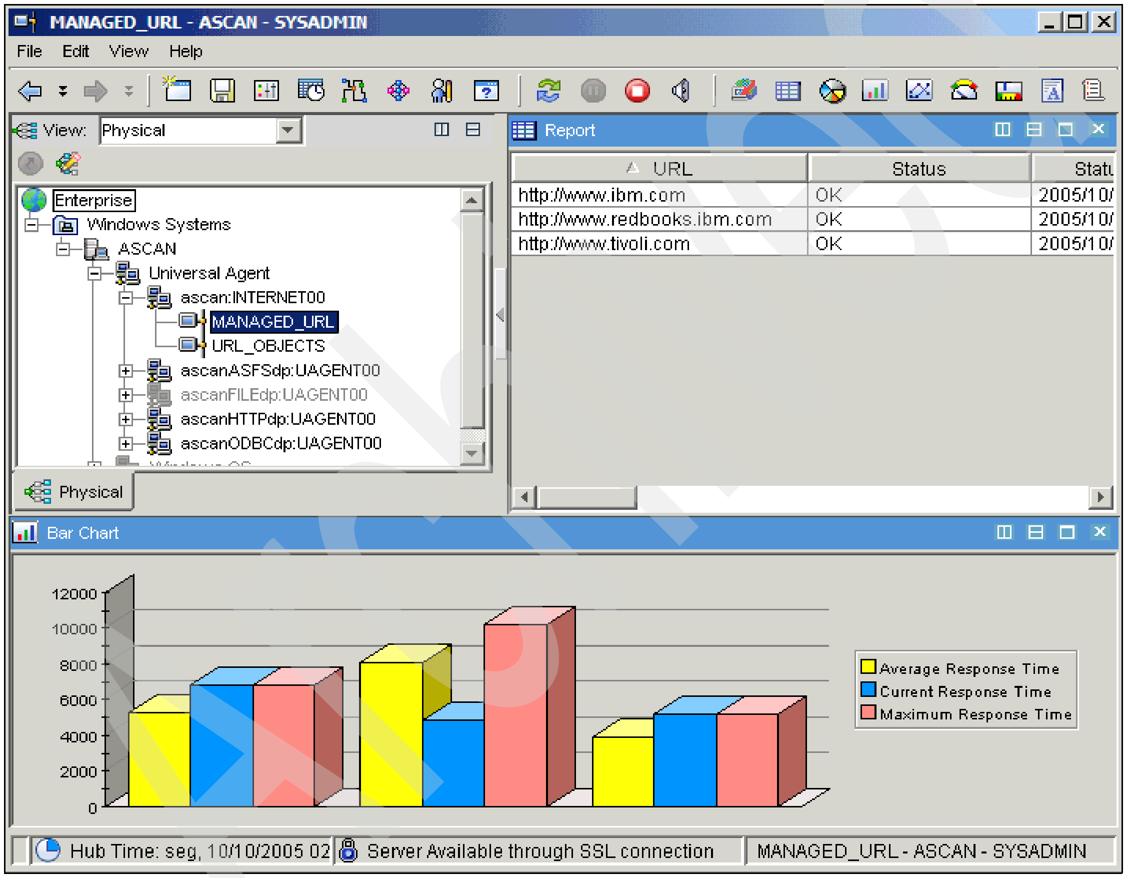
Next, configure some URLs to monitor with the HTTP data provider. To monitor a URL, first you must configure the KUMPULRS file located in the work directory and define the URLs that we want to monitor. The KUMPURLS is a definition file used to monitor URLs. If this file does not exist, we can only monitor URLs using situations or Take Action. Note that is not necessary to use the HTTP prefix.

Example 9-2 URLs monitoring definition file

```
* List of URLs to monitor by the Universal Agent HTTP Data Provider
www.ibm.com
http://www.tivoli.com * Tivoli Web Site
http://www.redbooks.ibm.com * Redbooks Web Site
```

Tip: Instead of directly editing the KUMPURLS file, you can also use **Take Action** → **URL Add**. This is the procedure described in the HTTP DP section of the *IBM Tivoli Universal Agent, Version 6.1.0, User's Guide,* SC32-9459. It has the added benefit of not requiring recycling of the Universal Agent for the new monitoring to take effect.

Start the Universal Agent. After a few minutes, the monitoring begins. Figure 9-6 shows the URL monitoring with a bar chart.



URL	Status	Statu
http://www.ibm.com	OK	2005/10/
http://www.redbooks.ibm.com	OK	2005/10/
http://www.tivoli.com	OK	2005/10/

Hub Time: seg, 10/10/2005 02 | Server Available through SSL connection | MANAGED_URL - ASCAN - SYSADMIN

Figure 9-6 URL monitoring in Tivoli Enterprise Portal

HTTP data provider Managed Systems

The HTTP data provider has the following Managed Systems: one for the INTERNET and one for HTTP data provider:

- For the INTERNET:

  `host-name:INTERNET00`

- For HTTP data provider:

  `host-nameHTTPdp:UAGENT00`

HTTP data provider URL attributes

The attributes shown in Table 9-5 are available to IBM Tivoli Monitoring situations URL monitoring and are displayed at Managed URLs table. Table 9-5 displays URL attributes.

Table 9-5 URL attributes

Attribute name	Type	Size	Description
Average Response Time	Integer	Long	The average observed managed URL response time in milliseconds.
Current Response Time	Integer	Long	The current observed managed URL response time in milliseconds.
HTTP Version	Character	8	The HTTP version (1.0 or 1.1) of the Web server for the target URL Web site.
ISP_Name	Character	68	The name of the Internet Service Provider (ISP).
Maximum Response Time	Integer	Long	The maximum observed managed URL response time in milliseconds.
Page Objects	Integer	Long	The total number of additional objects associated with the monitored page.
Page Size	Integer	Long	The page size, in bytes, of the received URL page.
Page Title	UTF-8	256	The page title of the received URL page.
Server Type	Character	64	The type of Web server used at the target URL Web site.
Status	Character	64	The current managed URL status (OK or status description).
Status Interval	Integer	Long	The elapsed time, in seconds, between status checks for the target URL.

Attribute name	Type	Size	Description
Status Timestamp	Character	32	The time when the current managed URL status was last taken.
Total Object Size	Integer	Long	The total number of bytes downloaded for the associated page objects.
Total Samples Taken	Integer	Long	The total number of samples taken for this URL since monitoring began.
URL	UTF-8	512	The target managed URL. You must use the format `http://`.
URL Alias	UTF-8	32	The user-specified alias for the URL.
User Name	UTF-8	32	The user ID that initiated monitoring for the target URL.

Table 9-6 lists the attributes for INTERNET table URL Objects, used to monitor availability and response time for embedded objects in the Web site.

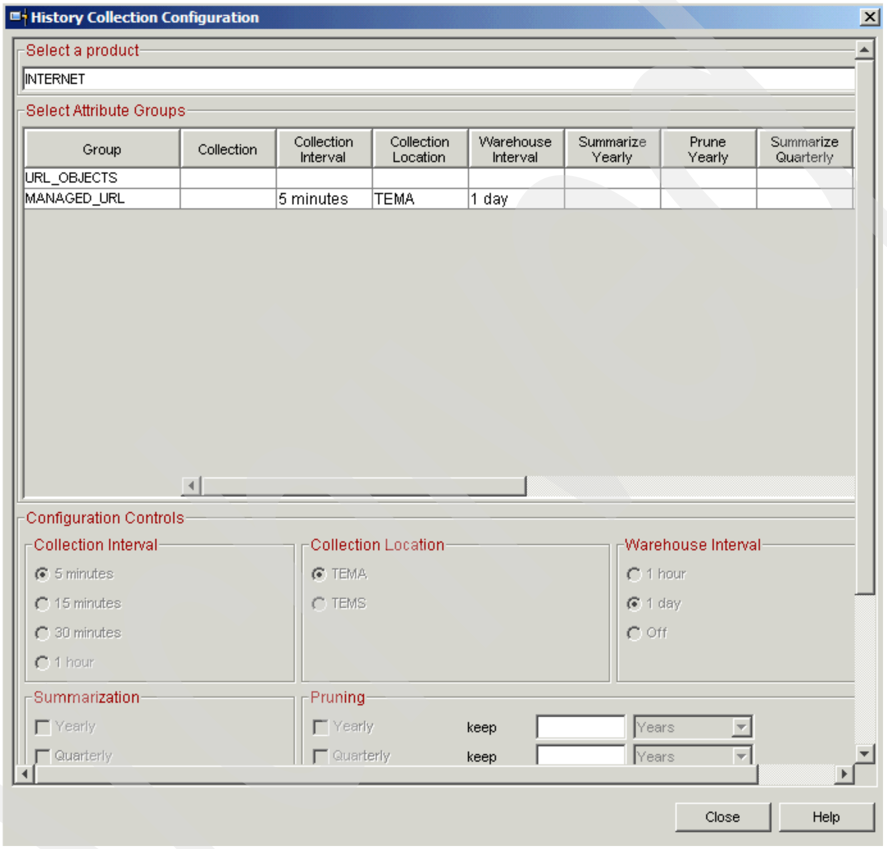
Table 9-6 URL objects

Attribute name	Type	Size	Description
Object Name	UTF-8	512	The name of the page object within target URL.
Object Size	Integer	Long	The size of page object within target URL.
URL	UTF-8	512	The target managed URL You must use the format, http://.

Historical Data configuration

Historical Data is configured to store data from URLs monitoring in the same way as others Historical Data are configured to store data to others agents.

To configure Historical Data: In the main window, click **History Configuration**. In the History Collection Configuration window, select **INTERNET** as the product.


History Collection Configuration

Select a product: INTERNET

Select Attribute Groups

Group	Collection	Collection Interval	Collection Location	Warehouse Interval	Summarize Yearly	Prune Yearly	Summarize Quarterly
URL_OBJECTS							
MANAGED_URL		5 minutes	TEMA	1 day			

Configuration Controls

Collection Interval: [✓ 5 minutes] 15 minutes 30 minutes 1 hour

Collection Location: [✓ TEMA] TEMS

Warehouse Interval: 1 hour [✓ 1 day] Off

Summarization: Yearly, Quarterly

Pruning: Yearly keep Years; Quarterly keep Years

Figure 9-7 History Collection Configuration window

To collect data about URL response time, we only have to configure the MANAGED_URL group.

Configuring situations

It is possible to create situations to be triggered for some conditions (for example if a URL is not available).

1. Right-click **MANAGED_URL** in the physical tree and choose **Situations** (Figure 9-8).

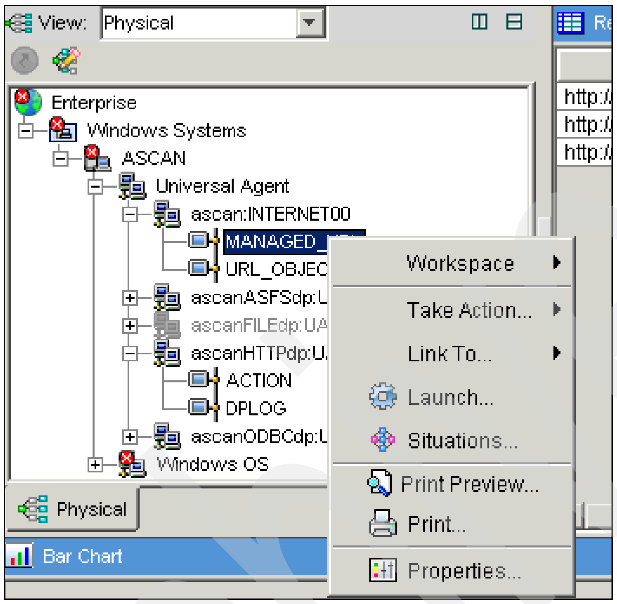


Figure 9-8 Situation configuration window

2. In the Situations for - MANAGED_URL window, right-click **MANAGED_URL** and select **Create New** (Figure 9-9).

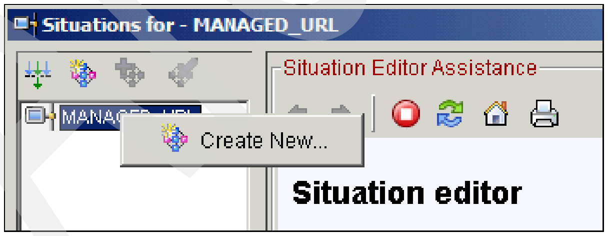
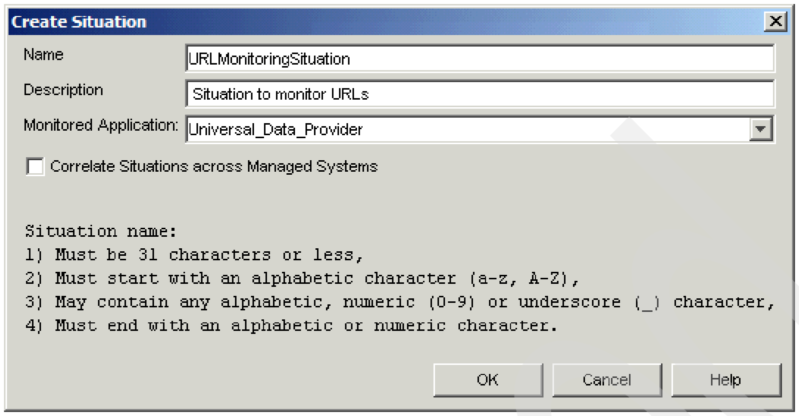


Figure 9-9 Create a new situation for HTTP data provider

3. Type a name and description for the Create Situation window (Figure 9-10).


Create Situation

Name: URLMonitoringSituation

Description: Situation to monitor URLs

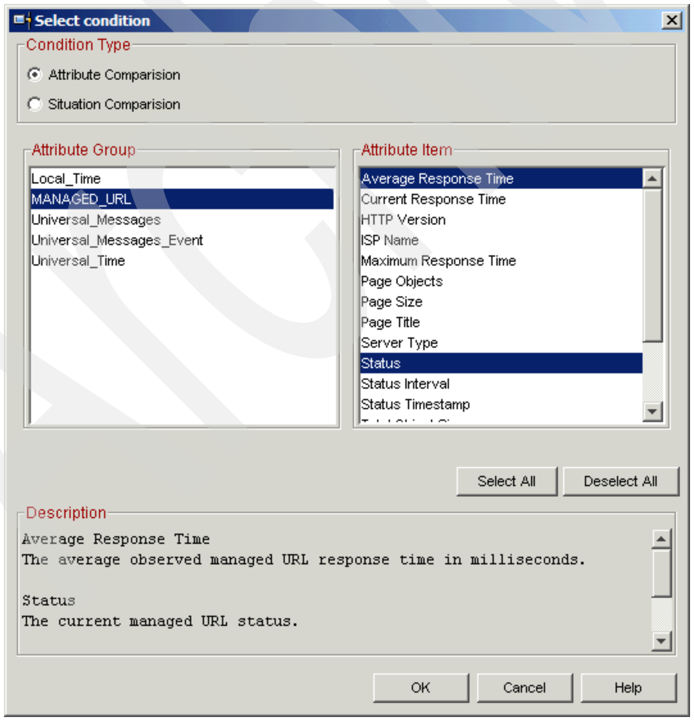
Monitored Application: Universal_Data_Provider

☐ Correlate Situations across Managed Systems

```
Situation name:
1) Must be 31 characters or less,
2) Must start with an alphabetic character (a-z, A-Z),
3) May contain any alphabetic, numeric (0-9) or underscore (_) character,
4) Must end with an alphabetic or numeric character.
```

Figure 9-10 Create Situation window

4. In the Select condition window, select **MANAGED_URL** from Attribute Group list, and from Attribute Item group select the items that we want to monitor: Average Response Time, Status and URL (Figure 9-11).


Select condition

Condition Type

● Attribute Comparision

○ Situation Comparision

Attribute Group: Local_Time, **MANAGED_URL** (selected), Universal_Messages, Universal_Messages_Event, Universal_Time

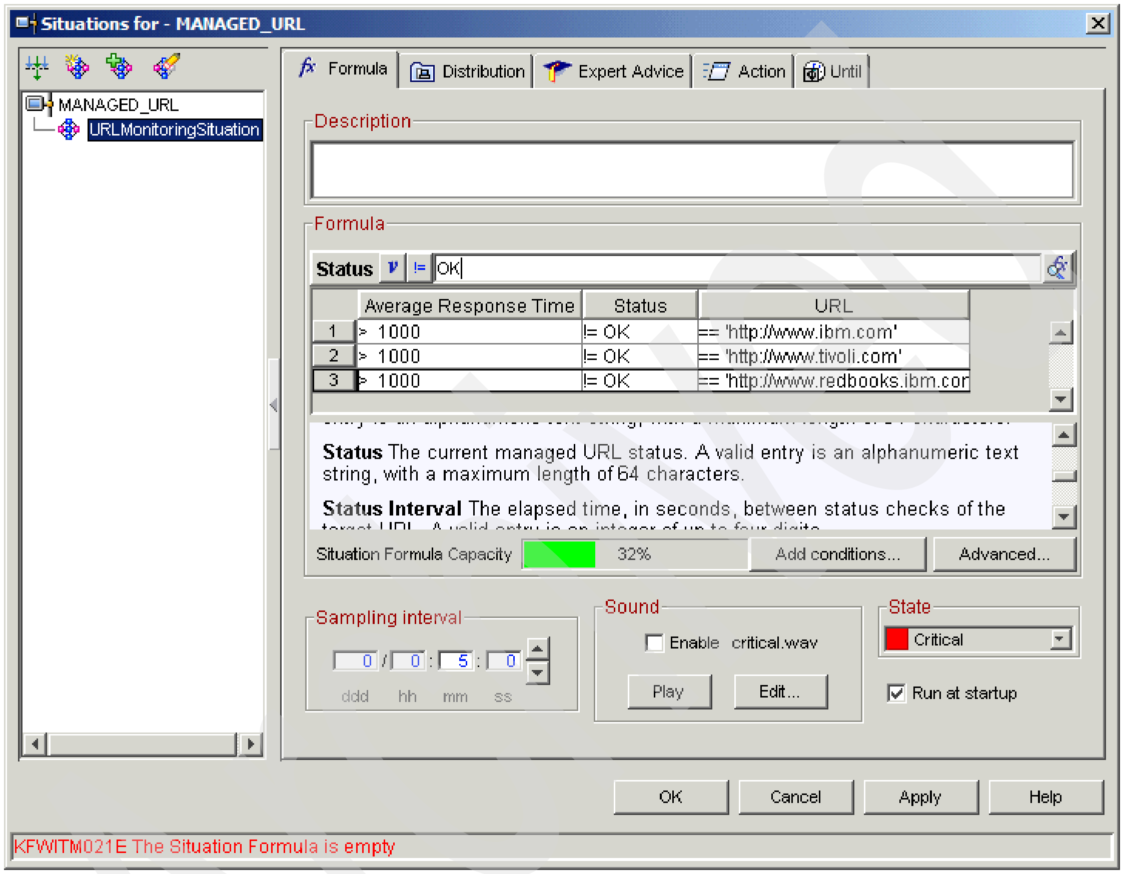
Attribute Item: **Average Response Time** (selected), Current Response Time, HTTP Version, ISP Name, Maximum Response Time, Page Objects, Page Size, Page Title, Server Type, **Status** (selected), Status Interval, Status Timestamp

Description

```
Average Response Time
The average observed managed URL response time in milliseconds.

Status
The current managed URL status.
```

Figure 9-11 Attribute items selection window

5. The last step is to configure some conditions to the situation be triggered. In this example, for the tree URLs, the situation was configured to alert when both URLs are not available (at the same time) or when the average response time is greater than 1000 ms in an interval of 5 minutes (Figure 9-12).



	Average Response Time	Status	URL
1	> 1000	!= OK	== 'http://www.ibm.com'
2	> 1000	!= OK	== 'http://www.tivoli.com'
3	> 1000	!= OK	== 'http://www.redbooks.ibm.cor

Status The current managed URL status. A valid entry is an alphanumeric text string, with a maximum length of 64 characters.

Status Interval The elapsed time, in seconds, between status checks of the

KFWITM021E The Situation Formula is empty

Figure 9-12 Situation parameters window

9.3.2 File data provider deployment scenario

The File data provider monitors data in sequential text files (for example, as log files). It reads the contents of files on the machine where the Universal Agent is installed or networked files via NFS. It is the simplest way to monitor data using the Universal Agent.

Starting the File data provider

We start a File data provider with the same method to start other data providers, configuring, for example, with the KUMA_STARTUP_DP parameter in the KUMENV file:

```
KUMA_STARTUP_DP=FILE
```

Note: A File data provider can be started also as a separated process using a command-line command to invoke the program: **`KUMPFILE`**.

Location

The File data provider must reside in the same host where we want to monitor a text file or on a remote workstation with a mapped logical drive.

Important: Remember that if a File data provider is monitoring a file in a remote system, the user ID or account associated with the File data provider must have authority to open and read the file in the remote system.

Frequency

The File data provider samples a text file for new records in a specific frequency. The frequency is determined as follows:

- For event type data, the File data provider samples data every 15 seconds or at the rate specified by the KUMP_DP_EVENT environment variable.
- For polled, sampled, and keyed data, the frequency is derived from time-to-live (TTL) value specified in the metafile divided by the sample factor. The default TTL is 300 seconds and the default sample factor is 5. The frequency for polled and sample data can be controlled using the KUMP_DP_SAMPLE_FACTOR environment variable.

Multiple record input

The File data provider supports multiple record inputs when multiple physical file records comprise one logical record. For example, the data for the two attributes can reside in one file record, and the data for a third attribute in another file record. This is when you would use the RECORDSET control statement in your metafile.

File name support

Applications can name output files based on several criteria such as day, week, or month. In these cases, we specify a monitoring file name pattern in the //SOURCE statement as:

```
//SOURCE FILE file-name-pattern-spec
```

In this case, the File data provider inspects all files in the designated path location, seeking files that match the defined pattern. The File data provider manages the most current matching file, based on whichever matching file has the highest number or date/time value. The appropriate file is determined by file name, instead of by file creation or modification date. The pound sign (#) defines the position of the numeric character in the file name. Table 9-7 lists file name pattern specification.

Table 9-7 File name pattern specification

File name pattern	Description
{########}.abc	Matches numeric file names of eight characters and the file extension *.abc*, such as 10252005.abc or 10262005.abc. File 10262005.abc is monitored because 10262005 is greater than 10252005.
{########}.*	Matches numeric file names of eight characters and ignores the file extension. Examples include 20051025.log, 20051101.log, and 10252005.abc. File 20051101.log is monitored because 20051101 is the largest number.
{######??}.abc	Matches numeric file names of eight characters and ignores the last two positions in the file name. Examples include 02110100.abc, 02110101.abc, and 02110202.abc. File 02110202.abc is monitored.
IN{######}.log	Matches file names starting with IN followed by six numerals and the file extension *.log*. Examples include IN021001.log, IN021002.log, and IN021004.log. File IN021004.log is monitored.
PS{###}FTP.txt	Matches file names starting with PS followed by three numerals, followed by FTP, and the extension *.txt*. Examples include PS001FTP.txt, PS005FTP.txt, and PS010FTP.txt. File PS010FTP.txt is monitored.

Some tips on specifying file name patterns:

- Use an asterisk (*) to ignore file extensions.
- If a specific file extension is defined, only files with the same extension are considered.
- Use braces {} to enclose the numeric part of the file name pattern.
- Use a pound sign to indicate each numeric element of a file name.
- Use a question mark to exclude each element of the naming convention that does not serve as search criteria in determining the appropriate file name.
- Use a dollar sign ($) to represent either any character or no character.

- The total number of pound signs and question marks enclosed in braces is significant. It must match the portion of file name exactly. For example, the pattern AA{####} instructs IBM Tivoli Universal Agent to look for file AA0001. File names, such as AA001 or AA00001, are not considered.
- The exact file name pattern, the constant and the numeric parts, must match the file name exactly.
- Wildcards are permitted. For example, if you want to match on both Log and LogA, you can specify Log{$}.
- To specify file names consisting of data component as year, month, and day use capital letters, such as Y, M and D. Table 9-8 lists the use of capital letters.

Table 9-8 Use of capital letters in file name pattern

Capital letters	Description
{YYYYMMDD}.log	Specifies candidate files with names such as 20050930.log or 20051015.log.
{MMDDYY}.log	Specifies files with names such as 101105.log or 110105.log.
{DDMMYYYY}.log	Specifies possible files with names such as 01092005.log. or 15082005.log.
MY{YYDDD}.log	Specifies files with names such as MY05202.log, MY05010.log or MY04350.log.

The File data provider checks for new files that match the defined pattern in the target location. The File data provider switches to a new file when a new file matches the defined pattern. It occurs when:

- The File Data Provider first starts up.
- The currently managed file no longer exists due to possible renaming or deleting.
- The existing file contents have changed.
- The check interval expired.

Note: The default interval is 10 minutes. You can change the interval to a longer or shorter interval value by specifying the environment variable KUMP_DP_FILE_SWITCH_CHECK_INTERVAL=seconds.

Data provider deployment

In this section we walk you through a File data provider scenario that monitors a NetView log (nv.log) to detect stop and start of a NetView server.

Metafile

Example 9-3 shows the metafile used in the File data provider deployment scenario to monitor the NetView log.

Example 9-3 Metafile example

```
//APPL nvlog
//NAME NV_LOG E
//SOURCE FILE c:\usr\ov\log\nv.log TAIL
//ATTRIBUTES
Date D 8
Time D 8
Source D 12 DLM='[]'
Message Z 2048
```

The first line is the application name. The second line is the name that the source will have. The third line is the data source file location, and the TAIL tells the Universal Agent to look for any additions to the file as they are happening. The fourth line tells the Universal Agent how attributes are separated in the file, in this case by space because there nothing follows the attribute command. The next two lines define eight characters for date and time and define them to be of time Display (D). In the log file, you can see that the source of the message is enclosed in square brackets. In this example, we want only the source name without the brackets, so we specify with the DLM parameter that the attribute is delimited by []. The last line has the Z data type and tells the Universal Agent that everything following from here until the end is the message portion and must be treated as a single attribute.

After the file is done, save it in the following location (this path can be changed using the KUMP_META_PATH environment variable):

- On Windows systems: C:\IBM\ITM\TMAITM6\metafiles
- On UNIX systems: $CANDLEHOME/$ARCH/um/metafiles

Note: By convention, every Universal agent metafile ends with the .mdl extension, but there are no restrictions. You can save the file with the name that you want.

Validating the metafile

Universal Agent has a validation program to check the metafile file. This command generates a report with the same name as the metafile, except it uses the .rpt extension. To run the validation program under the metafile, enter the command in Example 9-4.

Example 9-4 kumpcon validate command

```
C:\IBM\ITM\TMAITM6>kumpcon validate NVLOGMETAFILE.MDL
KUMPS001I Console input accepted.
KUMPV025I Processing input metafile NVLOGMETAFILE.MDL
KUMPV026I Processing record 0001 -> //APPL nvlog
KUMPV026I Processing record 0002 -> //NAME NV_LOG E
KUMPV026I Processing record 0003 -> //SOURCE FILE c:\usr\ov\log\nv.log TAIL
KUMPV026I Processing record 0004 -> //ATTRIBUTES
KUMPV026I Processing record 0005 -> Date D 8
KUMPV026I Processing record 0006 -> Time D 8
KUMPV026I Processing record 0007 -> Source D 12 DLM='[]'
KUMPV026I Processing record 0008 -> Message Z 2048
KUMPV000I Validation completed successfully
KUMPV090I Application metafile validation report saved in file
NVLOGMETAFILE.rpt.
KUMPS065I Do you wish to Import or Refresh this metafile?
<Import/Refresh/Cancel>
```

Example 9-5 shows a validation output file.

Example 9-5 Validation output report file

```
Application Name: nvlog; Definition Metafile Name:NVLOGMETAFILE.MDL
Attribute Group: NV_LOG
Type: Event data   Total Number of SOURCEs: 1
SOURCE is FILE c:\usr\ov\log\nv.log  (Tail mode)
Total Attributes: 4
Attribute delimiter is Space Character
Date    Display Type    Size 8    Delimiter is Space Character
Time    Display Type    Size 8    Delimiter is Space Character
Source Display Type    Size 12   Delimiter begin [ end ]
Message Last Type       Size 2048 Delimiter is Space Character
Total Attribute Groups: 1
```

Now tell the Universal Agent what to monitor. We do it in the following three ways:

- Configure the KUMPCNFG file, located in C:\IBM\ITM\TMAITM6\work, with the name of the metafile application and restart Universal Agent Service in the Manage Tivoli Enterprise Monitoring Services. For our example, nvlog:

```
* Universal Agent Configuration file
nvlog
```

- Activate metafile with console commands using IMPORT and REFRESH.
- Activate metafile with Take Action commands.

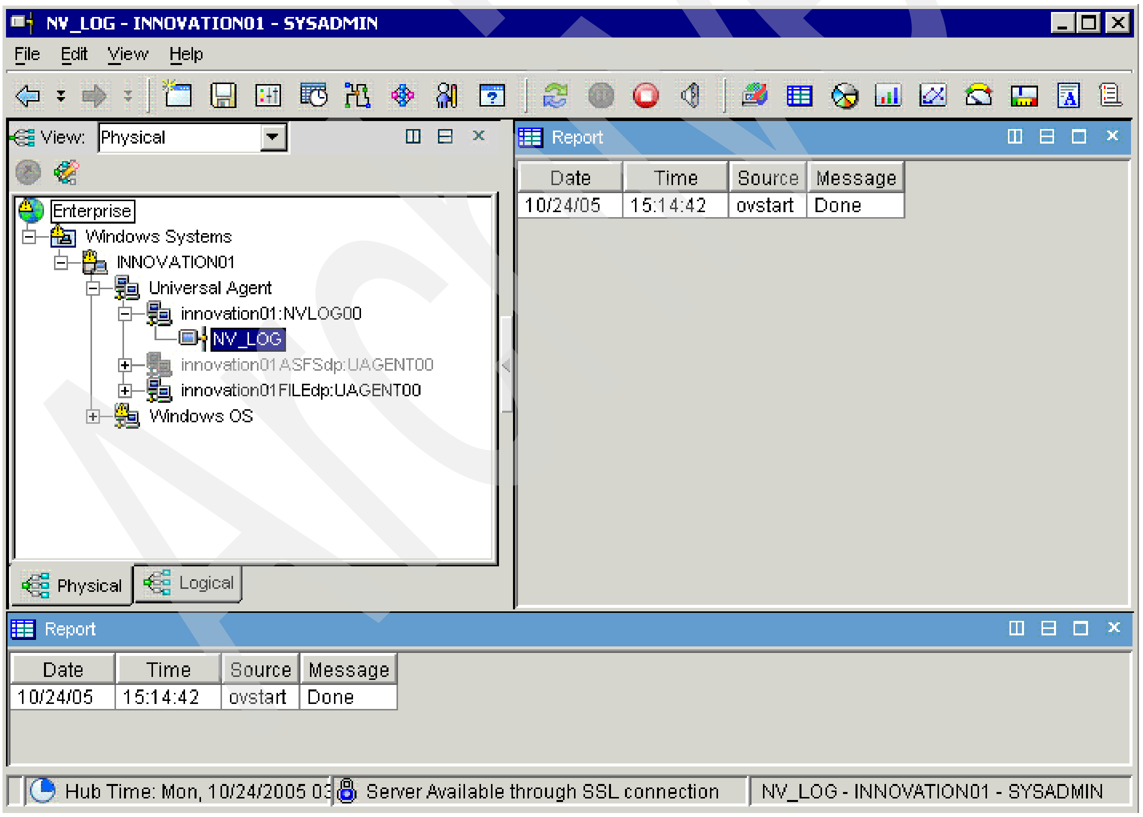
Console commands

In this example we activate the metafile with console commands. With this method we do not have to restart the IBM Tivoli Universal Agent. Example 9-6 shows the import of a metafile using the command line in Windows systems.

Example 9-6 Importing a metafile using the command line

```
C:\IBM\ITM\TMAITM6>kumpcon import NVLOGMETAFILE.MDL
KUMPS001I Console input accepted.
KUMPS020I Import successfully completed for NVLOGMETAFILE.MDL
```

Figure 9-13 shows the source NVLOG of the metafile after the import of the metafile application.



Date	Time	Source	Message
10/24/05	15:14:42	ovstart	Done

Hub Time: Mon, 10/24/2005 03 | Server Available through SSL connection | NV_LOG - INNOVATION01 - SYSADMIN

Figure 9-13 NVLOG metafile source

Create a situation

Now we create a situation to monitor the start and stop of the NetView server:

1. Right-click **NV_LOG** and select **Situations** (Figure 9-14).

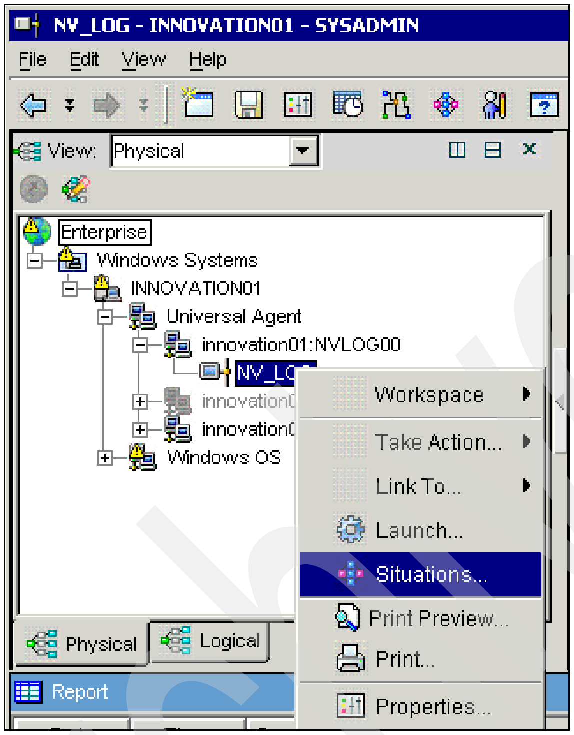
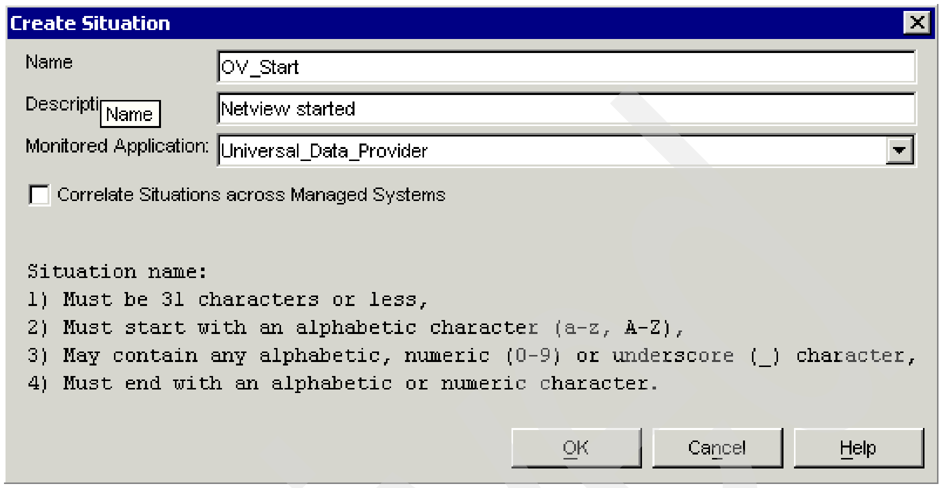


Figure 9-14 Creating a situation to monitor NetView start

2. In the **Create Situation** window, type a name and description for the situation (Figure 9-15).

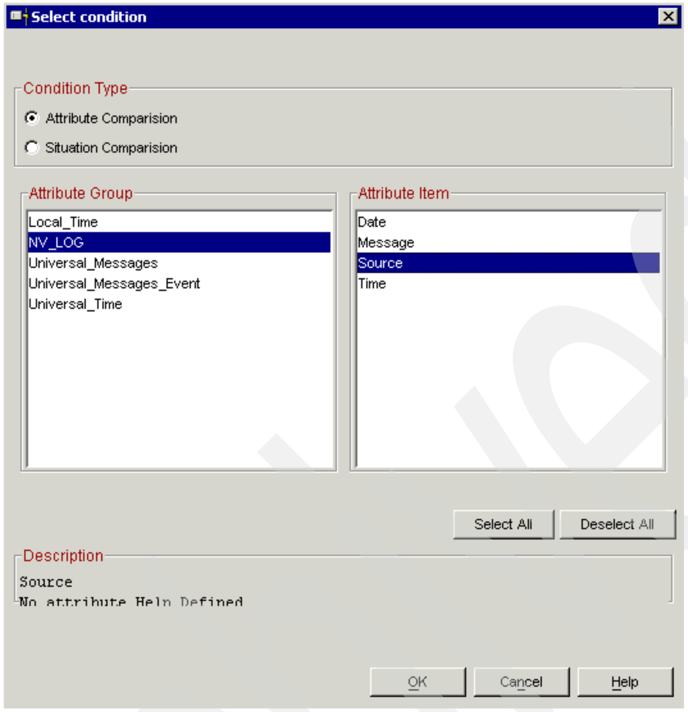


Name: OV_Start
Description: Netview started
Monitored Application: Universal_Data_Provider

Situation name:
1) Must be 31 characters or less,
2) Must start with an alphabetic character (a-z, A-Z),
3) May contain any alphabetic, numeric (0-9) or underscore (_) character,
4) Must end with an alphabetic or numeric character.

Figure 9-15 Type a name to identify the situation

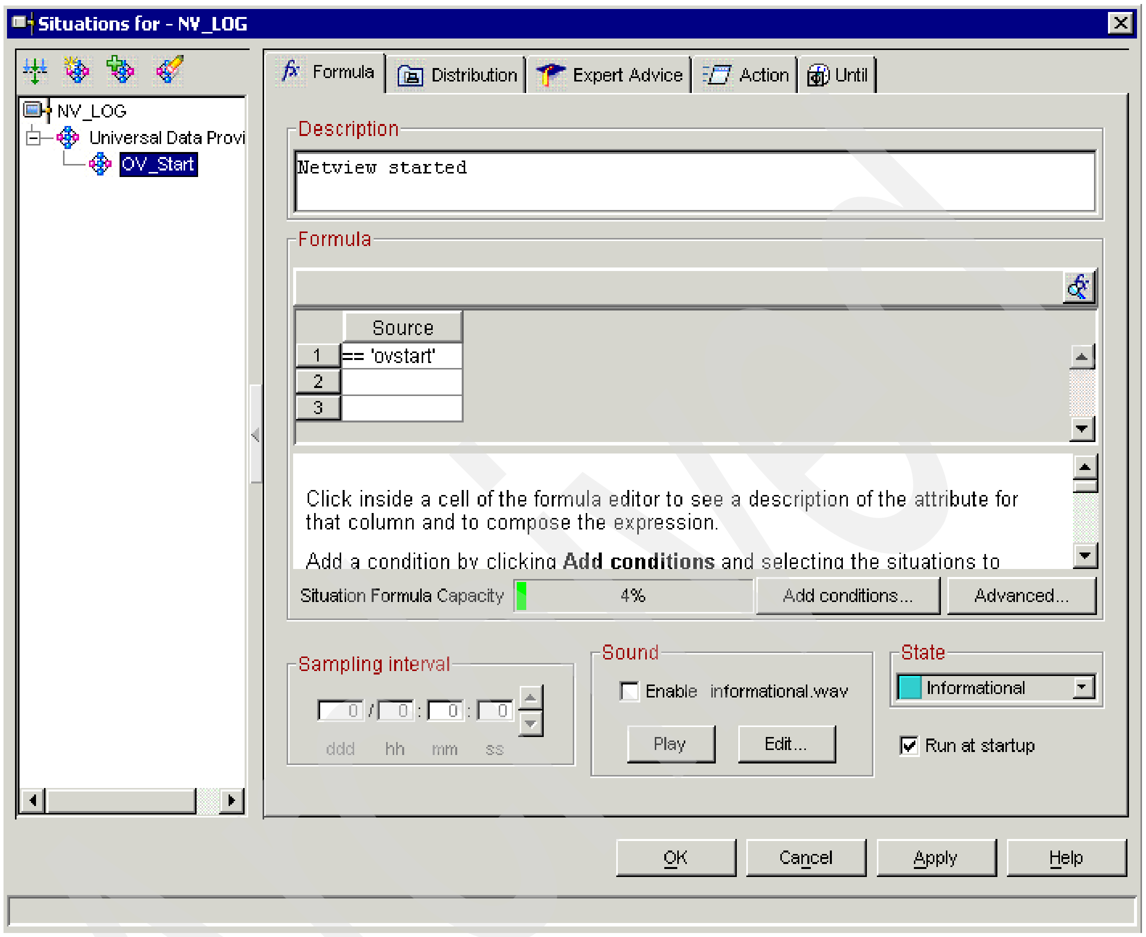
3. For the monitored attribute, select **Source** in the Attribute Item box (Figure 9-16).



```
Source
No attribute Help Defined
```

Figure 9-16 Select Source to be monitored by the situation

4. Be sure the situation has the value equal to **ovstart** and make the State **Informational** (Figure 9-17).



Description

Netview started

Formula

	Source
1	== 'ovstart'
2	
3	

Click inside a cell of the formula editor to see a description of the attribute for that column and to compose the expression.

Add a condition by clicking **Add conditions** and selecting the situations to

Situation Formula Capacity 4%

Sampling interval 0 / 0 : 0 : 0 ddd hh mm ss

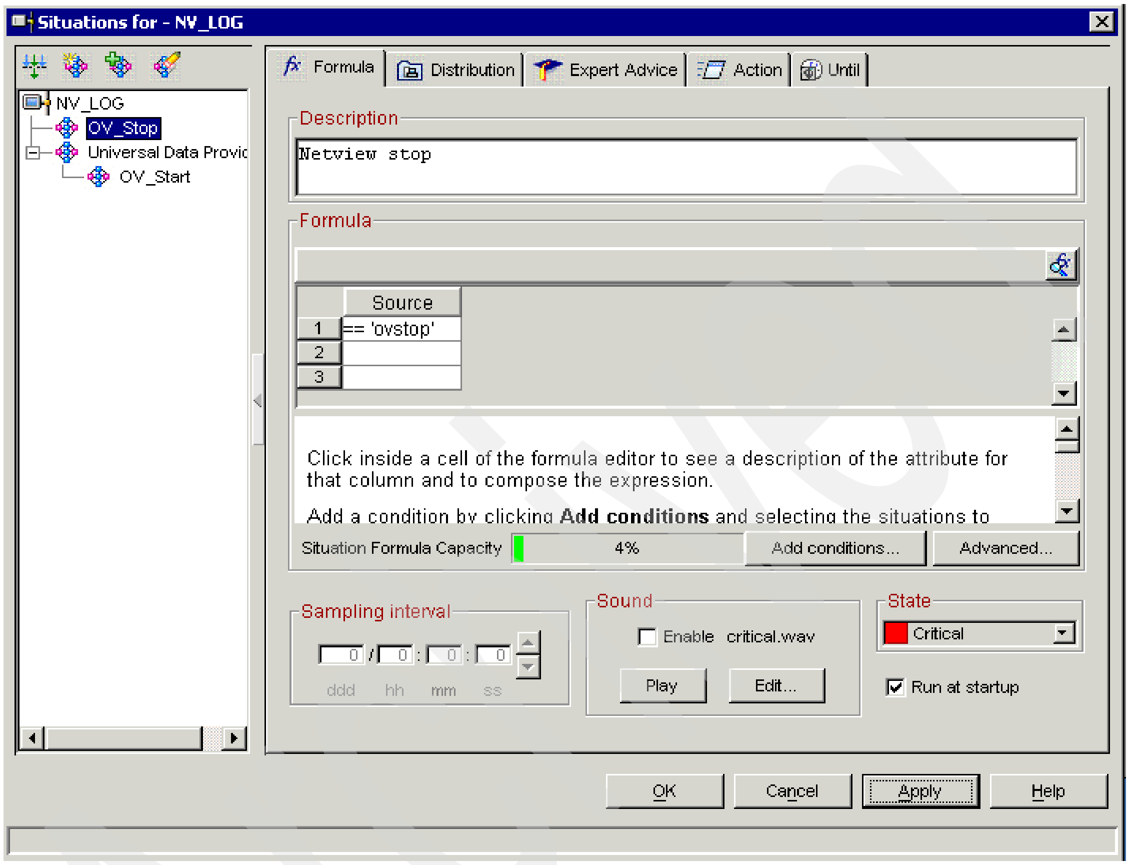
Sound ☐ Enable informational.wav

State Informational

☑ Run at startup

Figure 9-17 An informational situation to NetView start

5. Repeat steps 1 to 4 to create another situation, this time to monitor NetView stop. Call it **OV_Stop**, make sure the value is equal to **ovstop**, and make it a **Critical** situation (Figure 9-18).



Description

Netview stop

Formula

	Source
1	== 'ovstop'
2	
3	

Click inside a cell of the formula editor to see a description of the attribute for that column and to compose the expression.

Add a condition by clicking **Add conditions** and selecting the situations to

Situation Formula Capacity 4%

Sampling interval

0 / 0 : 0 : 0

ddd hh mm ss

Sound

☐ Enable critical.wav

State

🟥 Critical

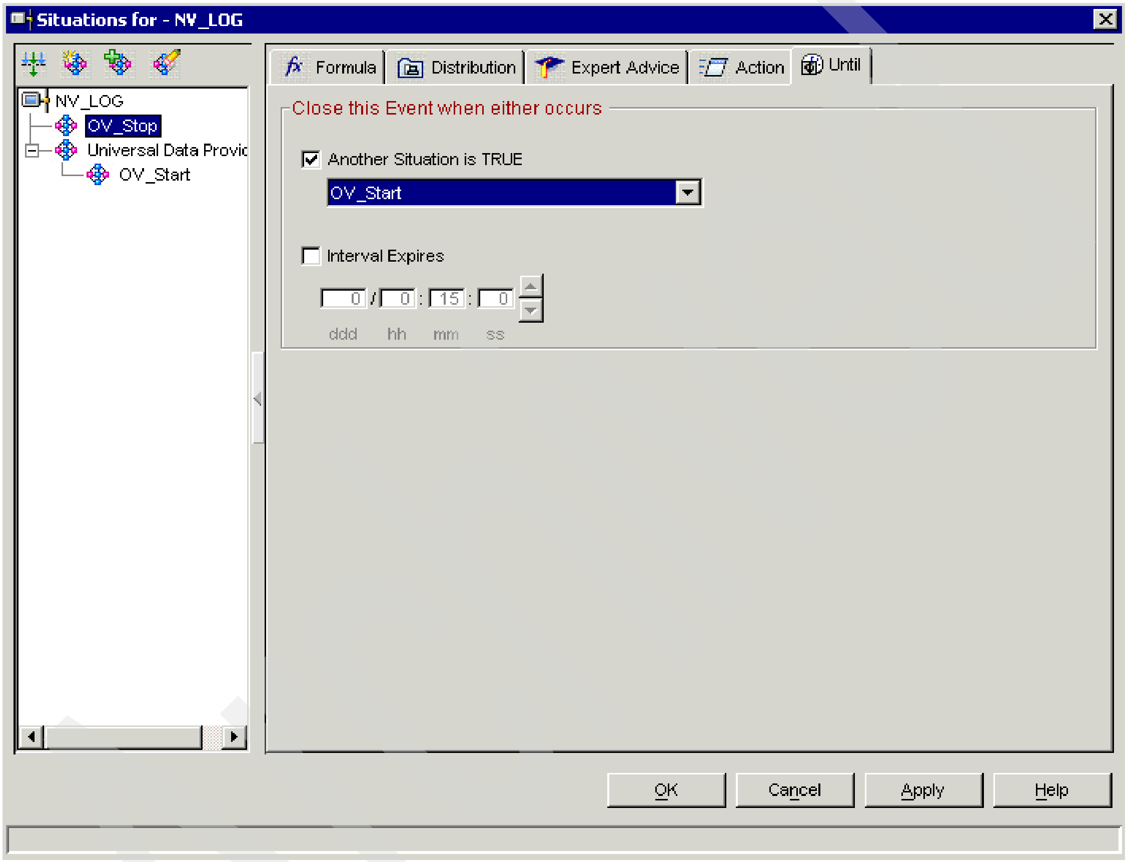
☑ Run at startup

Figure 9-18 Critical situation created to monitor NetView stop

Triggering the situations

Now, an event correlation between NetView start and NetView stop. This correlation closes automatically an `ovstop` event when NetView comes up.

1. Click the **Until** tab, check the **Another Situation is TRUE** box, and select the **OV_Start** situation (Figure 9-19).


Situations for - NV_LOG

NV_LOG
- OV_Stop
- Universal Data Provic
  - OV_Start

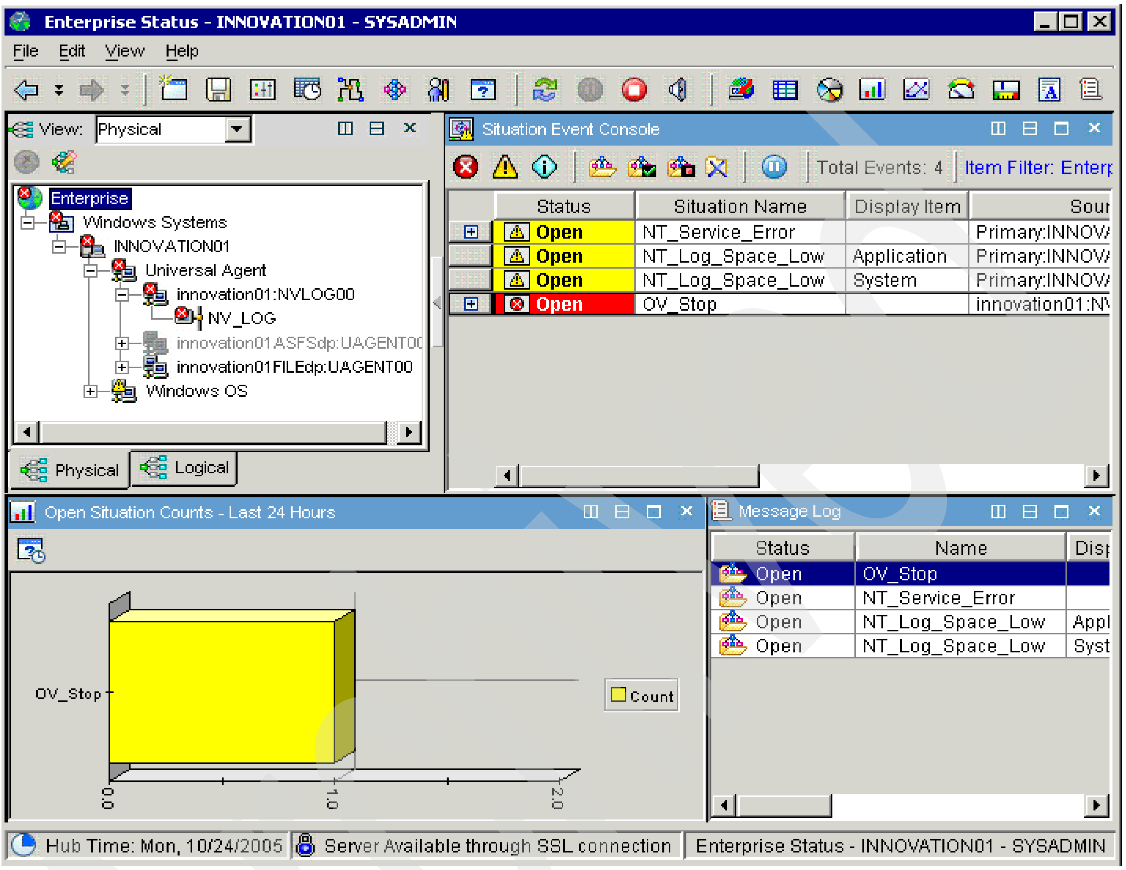
Close this Event when either occurs

[✓] Another Situation is TRUE: OV_Start

[ ] Interval Expires: 0 / 0 : 15 : 0 (ddd hh mm ss)

Figure 9-19 Correlation between situations

2. Trigger the situation by typing `ovstop` at the command prompt. Figure 9-20 shows the **`ovstop`** event in the Tivoli Enterprise Portal.



Enterprise
Windows Systems
INNOVATION01
Universal Agent
innovation01:NVLOG00
NV_LOG
innovation01ASFSdp:UAGENT00
innovation01FILEdp:UAGENT00
Windows OS

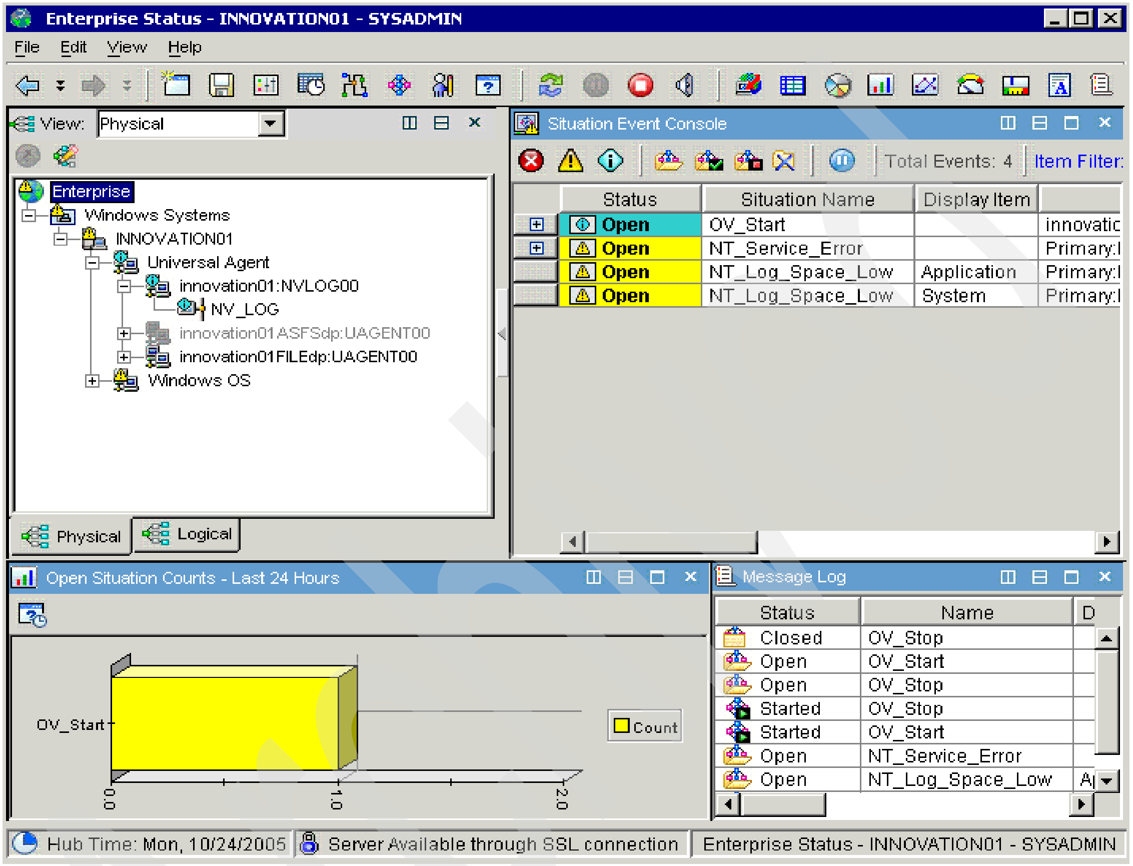
Status	Situation Name	Display Item	Sour
Open	NT_Service_Error		Primary:INNOVA
Open	NT_Log_Space_Low	Application	Primary:INNOVA
Open	NT_Log_Space_Low	System	Primary:INNOVA
Open	OV_Stop		innovation01:NV

Status	Name	Disp
Open	OV_Stop	
Open	NT_Service_Error	
Open	NT_Log_Space_Low	Appl
Open	NT_Log_Space_Low	Syst

Hub Time: Mon, 10/24/2005 | Server Available through SSL connection | Enterprise Status - INNOVATION01 - SYSADMIN

Figure 9-20 OV_Stop situation triggered in the Tivoli Enterprise Portal

3. To test the correlation type again in the command prompt the **ovstart** command, and after some minutes the OV_Stop event has been closed and the OV_Start event appears (Figure 9-21).



- Enterprise
  - Windows Systems
    - INNOVATION01
      - Universal Agent
        - innovation01:NVLOG00
          - NV_LOG
        - innovation01ASFSdp:UAGENT00
        - innovation01FILEdp:UAGENT00
      - Windows OS

Status	Situation Name	Display Item	
Open	OV_Start		innovatic
Open	NT_Service_Error		Primary:I
Open	NT_Log_Space_Low	Application	Primary:I
Open	NT_Log_Space_Low	System	Primary:I

Status	Name	D
Closed	OV_Stop	
Open	OV_Start	
Open	OV_Stop	
Started	OV_Stop	
Started	OV_Start	
Open	NT_Service_Error	
Open	NT_Log_Space_Low	A

Hub Time: Mon, 10/24/2005 | Server Available through SSL connection | Enterprise Status - INNOVATION01 - SYSADMIN

Figure 9-21 Event correlation between Ov_Stop and OV_Start situations

9.3.3 ODBC data provider deployment scenario

The ODBC is a standard application used to connect to relational databases for accessing data. Universal Agent enables you to run SQL statements and stored procedures in ODBC-compliant databases.

OBDC data provider runs as a separated data provider and is available only in Windows operating systems. The ODBC data provider runs on one machine and can simultaneously collect data from multiple remote databases.

Starting ODBC data provider

The ODBC data provider is started in the same way as other data providers, with the KUMA_STARTUP_DP parameter in the KUMENV file. For example:

```
KUMA_STARTUP_DP=ODBC
```

Note: The ODBC data provider can be started as a separate process by invoking:

```
KUMPODBC
```

Data can be collected as interval-driven or demand-driven. The default, demand-driven, signifies that data is collected only if a situation or report request is issued.

The GENERATE command

To create an ODBC metafile, you can use the **`generate`** command. Instead of creating the metafile manually, this command can save you time and effort by creating an ODBC metafile automatically. With this command, you also limit which tables are generated (user, system, view, or any combination). It does not support metafile generation for stored procedures. The **`generate`** command is available only in Windows operating systems.

Example 9-7 shows the correct syntax of the **`generate`** command.

Example 9-7 Generate command syntax

KUMPCON GENERATE *dataSourceName* user=*userid* pswd=*password*

Table 9-9 describes the parameters of the **`generate`** command.

Table 9-9 Generate command parameters

Parameter	Description
dataSourceName	Indicates the specific name of the configured data source that is used to create the ODBC metafile.
userid	The user ID that connects to the ODBC data source.
password	The password associated with the user ID connecting to the ODBC data source.

Data provider deployment

In this simple scenario, we want to know all users who are configured in our TEPS and are logged on to our TEPS database.

Generating the metafile

The first step is to generate the metafile used with the ODBC data provider; we use the **`generate`** command. Example 9-8 shows the use of the **`generate`** command to create a metafile to monitor TEPS2 database with ODBC data provider.

1. Type the following command to create the metafile:

   ```
   kumpcon generate teps2 user=db2admin pswd=db2admin
   ```

2. Type `1` on the prompt to select only user tables to include in the metafile and create a more targeted metafile.
3. On the next prompt, type `Y` to specify user tables.
4. Type `KFW` on the next prompt to select only tables beginning with KFW.

After this point the metafile is created.

Example 9-8 Metafile generation for TEPS2 database

```
C:\IBM\ITM\TMAITM6>kumpcon generate teps2 user=db2admin pswd=db2admin
KUMPS001I:Console input accepted.
ODBC Metafile Generation Utility
Indicate which type of tables to include in the generated metafile.
Select one or more of the following:
1) Include user tables
2) Include system tables
3) Include views
4) All of the above
Enter a number (or numbers) or type q to quit metafile generation.
If you enter more than one number, separate the numbers by a comma.
Type your selection(s) here:1
KUMPG031I:User tables will be included.
KUMPG003I:Using ODBC data source: teps2
KUMPG005I:Generating metafile: teps2.mdl
KUMPG038I:Do you want to pattern match on particular user tables? <Y/N>y
KUMPG041I:Specify beginning pattern matching characters for user tables:kfw
```

5. Edit the metafile and include `select` statements only for KFWLOGIN and KFWUSER tables. Example 9-9 shows the metafile after changes.

Example 9-9 Final version of the tesp2.mdl metafile

```
//APPL teps2
//NAME KFWLOGIN S 300
//SOURCE ODBC teps2 user=db2admin pswd=db2admin
//SQL Select * from TEPS.KFWLOGIN
//ATTRIBUTES
TIMEON  D  28
USERID  D  32
```

```
IPADDR  D  32
IOR     D  256
//NAME KFWUSER S 300
//SOURCE ODBC teps2 user=db2admin pswd=db2admin
//SQL Select * from TEPS.KFWUSER
//ATTRIBUTES
ID          D  32
NAME        D  48
TEXT        D  64
AFFINITIES  D  44
AUTH        C  999999
AUTHEX      D  64
LSTDATE     D  16
LSTUSRPRF   D  32
```

Validating the metafile

Before importing the metafile, it is necessary to validate it. Enter the following command:

```
kumpcon validate teps2.mdl
```

Example 9-10 shows the **`kumpcon validate`** output.

Example 9-10 TEPS2.mdl metafile validation

```
C:\IBM\ITM\TMAITM6>kumpcon validate teps2.mdl
KUMPS001I:Console input accepted.
KUMPV025I:Processing input metafile teps2.mdl
KUMPV026I:Processing record 0001 -> //APPL teps2
KUMPV026I:Processing record 0002 -> //NAME KFWLOGIN S 300
KUMPV026I:Processing record 0003 -> //SOURCE ODBC teps2 user=db2admin
pswd=db2admin
KUMPV026I:Processing record 0004 -> //SQL Select * from TEPS.KFWLOGIN
KUMPV026I:Processing record 0005 -> //ATTRIBUTES
KUMPV026I:Processing record 0006 -> TIMEON  D  28
KUMPV026I:Processing record 0007 -> USERID  D  32
KUMPV026I:Processing record 0008 -> IPADDR  D  32
KUMPV026I:Processing record 0009 -> IOR     D  256
KUMPV026I:Processing record 0010 -> //NAME KFWUSER S 300
KUMPV026I:Processing record 0011 -> //SOURCE ODBC teps2 user=db2admin
pswd=db2admin
KUMPV026I:Processing record 0012 -> //SQL Select * from TEPS.KFWUSER
KUMPV026I:Processing record 0013 -> //ATTRIBUTES
KUMPV026I:Processing record 0014 -> ID          D  32
KUMPV026I:Processing record 0015 -> NAME        D  48
KUMPV026I:Processing record 0016 -> TEXT        D  64
KUMPV026I:Processing record 0017 -> AFFINITIES  D  44
KUMPV026I:Processing record 0018 -> AUTH        C  999999
KUMPV026I:Processing record 0019 -> AUTHEX      D  64
```

```
KUMPV026I:Processing record 0020 -> LSTDATE     D  16
KUMPV026I:Processing record 0021 -> LSTUSRPRF   D  32
KUMPV000I:Validation completed successfully
KUMPV090I:Application metafile validation report saved in file teps2.rpt.
KUMPS065I:Do you wish to Import or Refresh this metafile?
<Import/Refresh/Cancel>
```

Note: Type `Cancel` on the prompt at the end of the **`validate`** command. In this example we use Take Action commands to import the metafile.

The **`validate`** command generates a report (same name as the metafile, but with the .rpt extension) showing the application definition. Example 9-11 shows the output report for teps2.mdl metafile.

Example 9-11 Output report for TEPS2.mdl metafile

```
Application Name: teps2; Definition Metafile Name: TEPS2.MDL
Attribute Group: KFWLOGIN
Type: Sample data   TTL: 300 seconds   Total Number of SOURCEs: 1
SOURCE is ODBC
Data source: teps2
Userid: db2admin
SQL statement: SELECT * from TEPS.KFWLOGIN
Total Attributes: 4
Attribute delimiter is Space Character
TIMEON  Display Type    Size 28   Delimiter is Space Character
USERID  Display Type    Size 32   Delimiter is Space Character
IPADDR  Display Type    Size 32   Delimiter is Space Character
IOR     Display Type    Size 256  Delimiter is Space Character
Attribute Group: KFWUSER
Type: Sample data   TTL: 300 seconds   Total Number of SOURCEs: 1
SOURCE is ODBC
Data source: teps2
Userid: db2admin
SQL statement: SELECT * from TEPS.KFWUSER
Total Attributes: 8
Attribute delimiter is Space Character
ID         Display Type    Size 32   Delimiter is Space Character
NAME       Display Type    Size 48   Delimiter is Space Character
TEXT       Display Type    Size 64   Delimiter is Space Character
AFFINITIES Display Type    Size 44   Delimiter is Space Character
AUTH       Counter Type    Size 4    Delimiter is Space Character
AUTHEX     Display Type    Size 64   Delimiter is Space Character
LSTDATE    Display Type    Size 16   Delimiter is Space Character
LSTUSRPRF  Display Type    Size 32   Delimiter is Space Character
Total Attribute Groups: 2
```

Importing metafile with Take Action command

In this scenario we use a Take Action command in the Tivoli Enterprise Portal to import the metafile:

1. Right-click on the ODBC data provider Managed System, and select **Take Action** → **Select** (Figure 9-22).

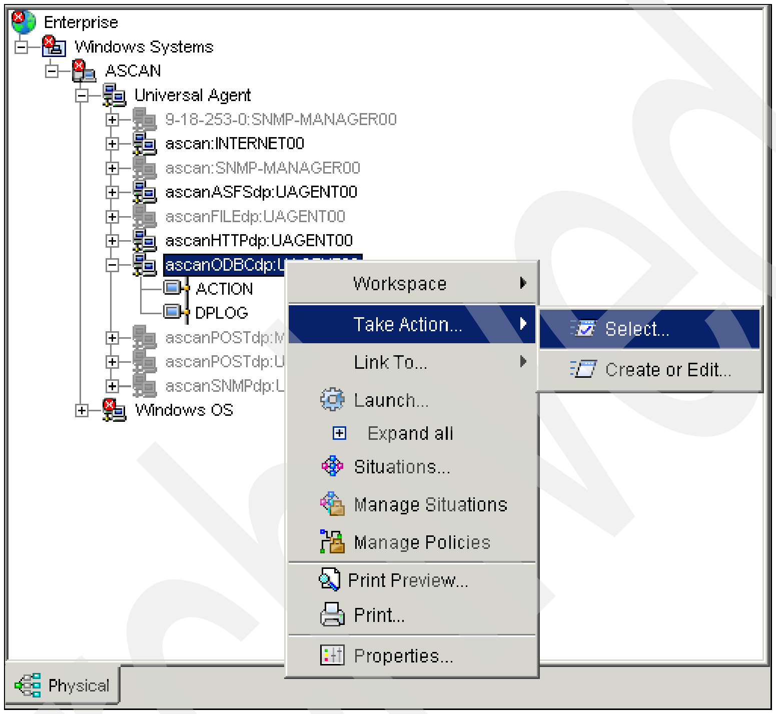


Figure 9-22 Importing ODBC metafile using Take Action command

2. In the Take Action window, select **Control Import** (Figure 9-23).

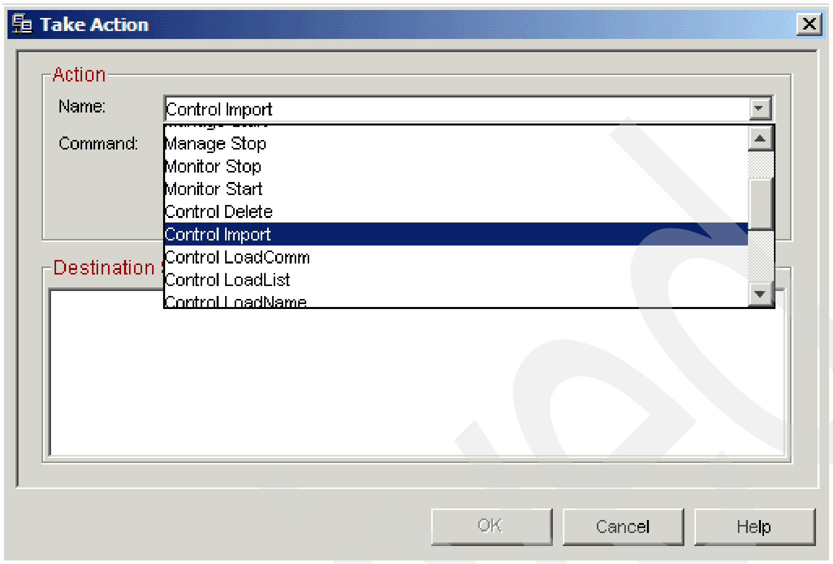
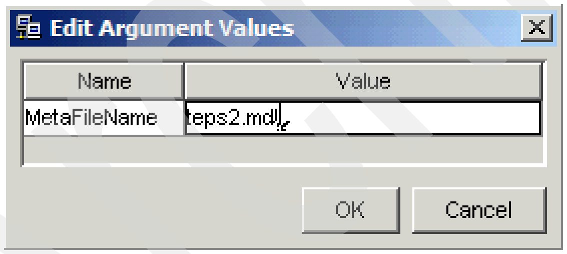


Figure 9-23 Selecting the Control Import action in the Take Action window

3. In the Edit Arguments Values window (Figure 9-24), type the name of the ODBC data provider metafile of this scenario, in this case `teps2.mdl`.



Name	Value
MetaFileName	teps2.mdl

Figure 9-24 Defining the metafile to import

4. Select the Managed System to execute the action and click **OK** (Figure 9-25).

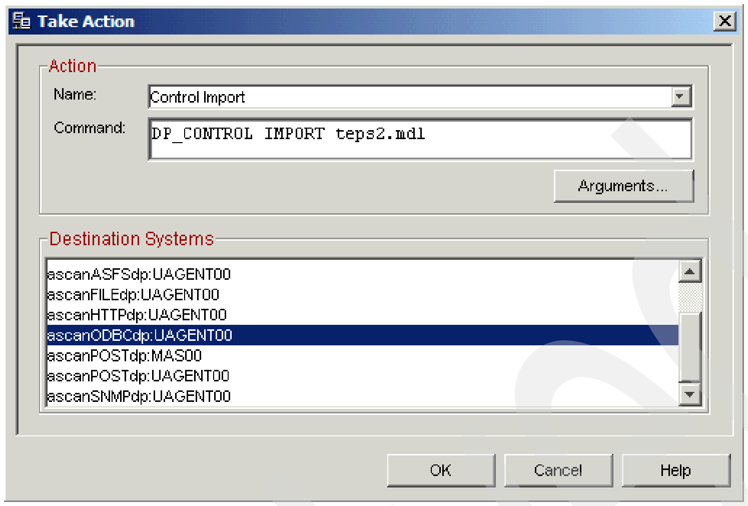
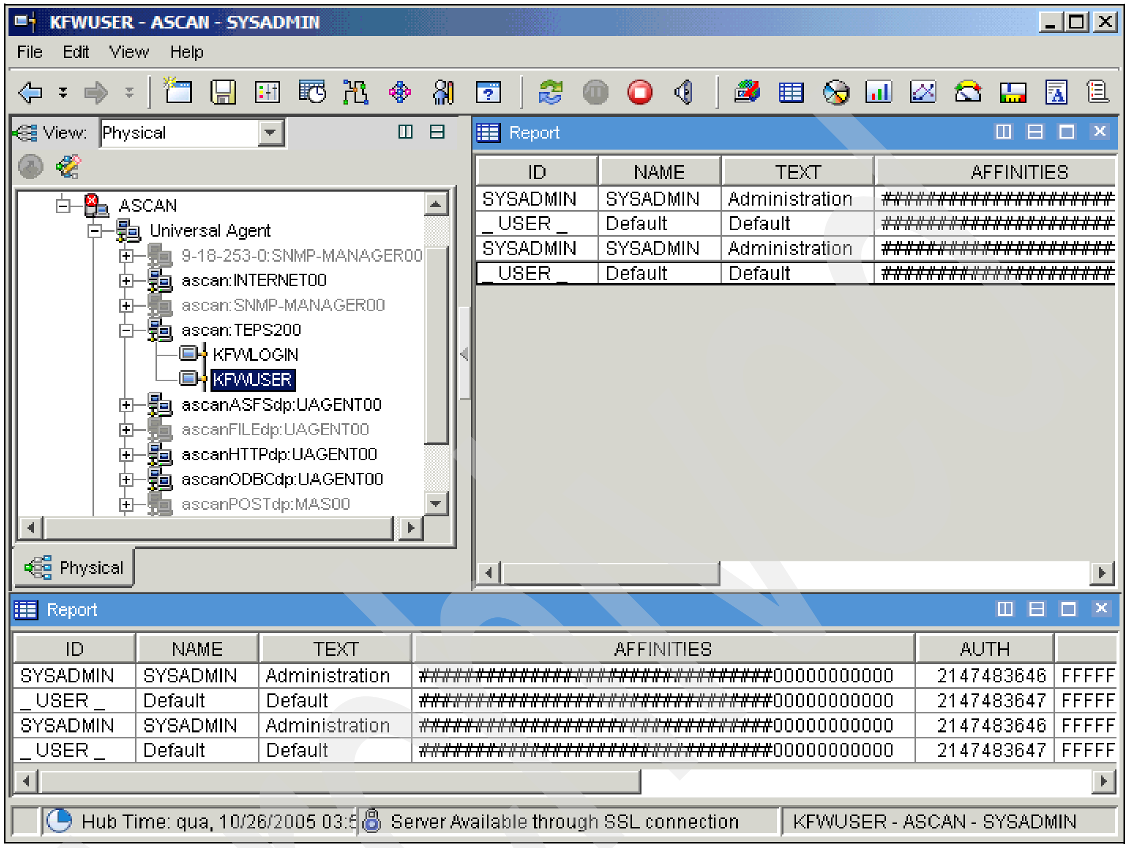


Figure 9-25 Selecting the Managed System to execute the action

5. Figure 9-26 shows the new application of the metafile with two attribute groups (KFWLOGIN and KFWUSER) after the import.



ID	NAME	TEXT	AFFINITIES
SYSADMIN	SYSADMIN	Administration	#######################
_ USER _	Default	Default	#######################
SYSADMIN	SYSADMIN	Administration	#######################
_ USER _	Default	Default	#######################

Report

ID	NAME	TEXT	AFFINITIES	AUTH	
SYSADMIN	SYSADMIN	Administration	####################################00000000000	2147483646	FFFFF
_ USER _	Default	Default	####################################00000000000	2147483647	FFFFF
SYSADMIN	SYSADMIN	Administration	####################################00000000000	2147483646	FFFFF
_ USER _	Default	Default	####################################00000000000	2147483647	FFFFF

Hub Time: qua, 10/26/2005 03:5 | Server Available through SSL connection | KFWUSER - ASCAN - SYSADMIN

Figure 9-26   New attribute groups with rows with the same data

Note that there are multiple rows with the same data. To prevent this, replace the statement S with K in the //NAME parameter. The K statement indicates that the table is a keyed table, preventing the same retrieved rows from being added multiple times whenever the SQL Select statement is started. Example 9-12 shows the metafile after the changes in the //NAME parameter.

Example 9-12   //NAME parameter with the correct statement

```
//APPL teps2
//NAME KFWLOGIN K 300
//SOURCE ODBC teps2 user=db2admin pswd=db2admin
//SQL Select * from TEPS.KFWLOGIN
//ATTRIBUTES
TIMEON  D  28
```

```
USERID  D  32
IPADDR  D  32
IOR     D  256
//NAME KFWUSER K 300
//SOURCE ODBC teps2 user=db2admin pswd=db2admin
//SQL Select * from TEPS.KFWUSER
//ATTRIBUTES
ID          D  32
NAME        D  48
TEXT        D  64
AFFINITIES  D  44
AUTH        C  999999
AUTHEX      D  64
LSTDATE     D  16
LSTUSRPRF   D  32
```

After changing the statement and saving the metafile, you must refresh it in the Universal Agent. Enter the following command to refresh the metafile:

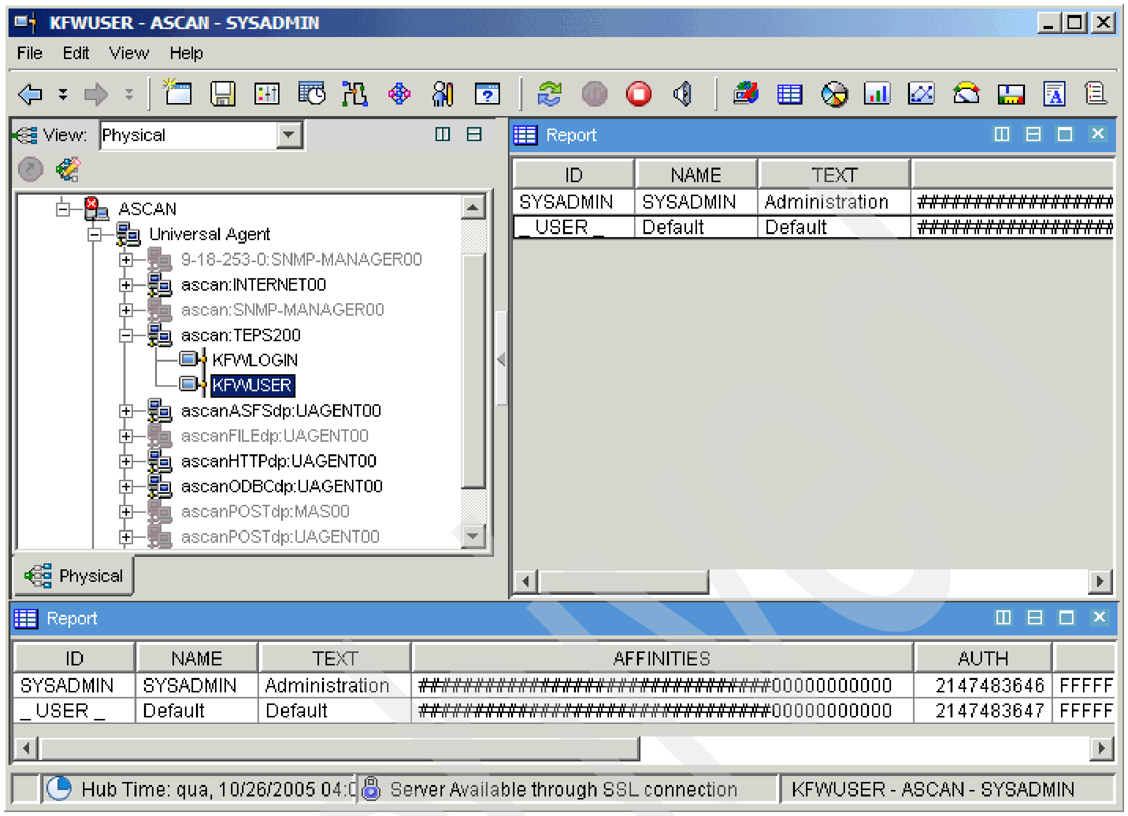
```
kumpcon refresh teps2.mdl
```

Example 9-13 shows the output of the **refresh** command on the teps2.mdl metafile.

Example 9-13 Kumpcon refresh output

```
C:\IBM\ITM\TMAITM6>kumpcon refresh teps2.mdl
KUMPS001I:Console input accepted.
KUMPS084I:Selecting ODBC DP based on metafile type
KUMPS025I:Confirm <Yes/No> to refresh teps2.mdl
yes
KUMPS027I:Refresh successful.
```

Now the data in the Tivoli Enterprise Portal is correct (Figure 9-27 on page 576).



ID	NAME	TEXT	
SYSADMIN	SYSADMIN	Administration	##################
_ USER _	Default	Default	##################

ID	NAME	TEXT	AFFINITIES	AUTH	
SYSADMIN	SYSADMIN	Administration	################################00000000000	2147483646	FFFFF
_ USER _	Default	Default	################################00000000000	2147483647	FFFFF

Hub Time: qua, 10/26/2005 04:0
Server Available through SSL connection
KFWUSER - ASCAN - SYSADMIN

Figure 9-27 Data in Tivoli Enterprise Portal after changes in the metafile

9.4 IBM Tivoli Universal Agent troubleshooting

In the following section, we cover some troubleshooting techniques for the IBM Tivoli Universal Agent.

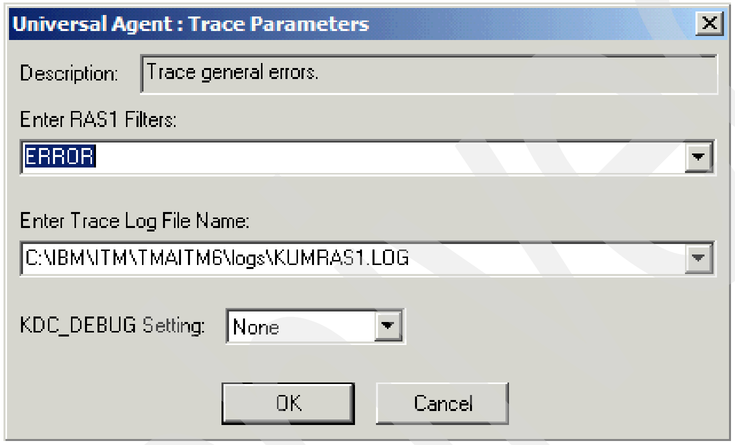
9.4.1 Setting the trace

The IBM Tivoli Universal Agent uses the RAS1 trace that is written in the logs subdirectory. By default, RAS1 trace has trace level ERROR. We set tracing options for Universal Agent in KUMENV file in Windows and um.ini in UNIX systems. In Windows systems, the log file is located in C:\IBM\ITM\TMAITM6\logs\KUMRAS1.LOG.

Note: Detailed RAS1 tracing might degrade the Universal Agent performance due to high CPU usage and I/O overhead.

Setting the IBM Tivoli Universal Agent trace involves the following steps:

1. In the Manage Tivoli Enterprise Monitoring Services window, right-click **Universal Agent**, then select **Advanced** → **Edit Trace Parms**.
2. In the Universal Agent: Trace Parameters window, choose the appropriate filter for the trace log (Figure 9-28).



Description: Trace general errors.

Enter RAS1 Filters: ERROR

Enter Trace Log File Name: C:\IBM\ITM\TMAITM6\logs\KUMRAS1.LOG

KDC_DEBUG Setting: None

Figure 9-28 Setting trace parameters

3. Restart the Universal Agent to implement changes.

9.4.2 UAGENT application

The UAGENT application is a diagnostic tool that comes online during data provider startup, and comes with the DPLOG and ACTION workspaces. This application helps determine problems with Universal Agent.

DPLOG

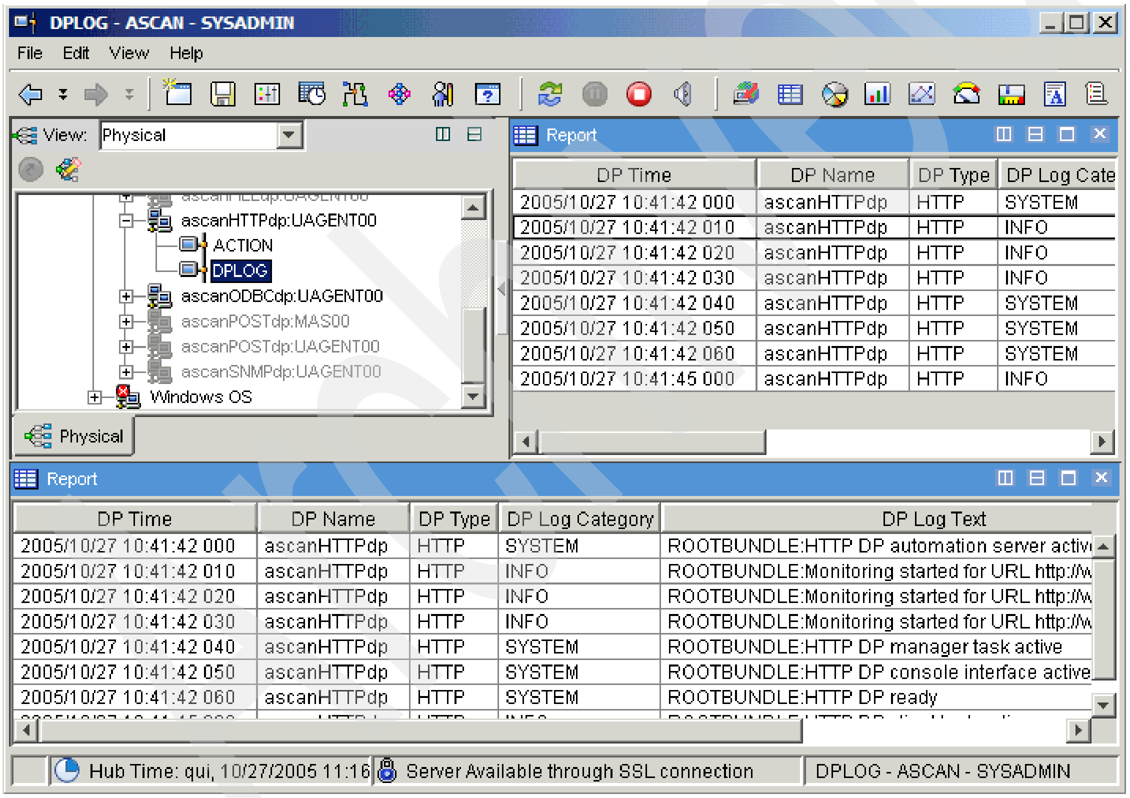
DPLOG is an event table that maintains the most recent 100 rows, unless we change it with the KUMA_MAX_EVENT_ENTRIES environment variable. It shows informational and error messages about data providers. Information in this table includes:

- Whether a metafile has been validated successfully, or if it failed validation (which means that its application will not come online)
- Whether a data source was available at startup

- Which console ports and socket listening ports were used or unavailable
- When monitoring started and stopped for a data source
- When monitoring switched from one file to another
- When an API or socket client program got connected and disconnected

ACTION

The Action table rows have a time-to-live value of 30 minutes. ACTION, different from DPLOG table, is shared by all data providers. The ACTION table under every UAGENT application has the same rows, and it indicates what happened to a specific Take Action command. Figure 9-29 shows the DPLOG workspace for HTTP UAGENT application.



DP Time	DP Name	DP Type	DP Log Cate
2005/10/27 10:41:42 000	ascanHTTPdp	HTTP	SYSTEM
2005/10/27 10:41:42 010	ascanHTTPdp	HTTP	INFO
2005/10/27 10:41:42 020	ascanHTTPdp	HTTP	INFO
2005/10/27 10:41:42 030	ascanHTTPdp	HTTP	INFO
2005/10/27 10:41:42 040	ascanHTTPdp	HTTP	SYSTEM
2005/10/27 10:41:42 050	ascanHTTPdp	HTTP	SYSTEM
2005/10/27 10:41:42 060	ascanHTTPdp	HTTP	SYSTEM
2005/10/27 10:41:45 000	ascanHTTPdp	HTTP	INFO

DP Time	DP Name	DP Type	DP Log Category	DP Log Text
2005/10/27 10:41:42 000	ascanHTTPdp	HTTP	SYSTEM	ROOTBUNDLE:HTTP DP automation server activ
2005/10/27 10:41:42 010	ascanHTTPdp	HTTP	INFO	ROOTBUNDLE:Monitoring started for URL http://w
2005/10/27 10:41:42 020	ascanHTTPdp	HTTP	INFO	ROOTBUNDLE:Monitoring started for URL http://w
2005/10/27 10:41:42 030	ascanHTTPdp	HTTP	INFO	ROOTBUNDLE:Monitoring started for URL http://w
2005/10/27 10:41:42 040	ascanHTTPdp	HTTP	SYSTEM	ROOTBUNDLE:HTTP DP manager task active
2005/10/27 10:41:42 050	ascanHTTPdp	HTTP	SYSTEM	ROOTBUNDLE:HTTP DP console interface active
2005/10/27 10:41:42 060	ascanHTTPdp	HTTP	SYSTEM	ROOTBUNDLE:HTTP DP ready

Hub Time: qui, 10/27/2005 11:16 | Server Available through SSL connection | DPLOG - ASCAN - SYSADMIN

Figure 9-29 DPLOG workspace

The two most common Universal Agent problems are:

- One or more managed systems do not come online.
- The managed systems are online but the workspaces are empty.

9.4.3 IBM Tivoli Universal Agent problem determination

Some IBM Tivoli Monitoring problems are related to application data definition, environment variables, Tivoli Enterprise Monitoring Server, and Tivoli Enterprise Portal configuration.

Therefore, begin the problem determination with data providers, then proceed with the other Tivoli Monitoring Services components.

Tips to determine problems with Universal Agent include:

- Validate the metafiles using the console command VALIDATE. Review the validation messages and report. Resolve all identified errors and warnings.
- Verify that the first three characters of the application name defined in the APPL statement of the metafile are unique throughout the enterprise.
- Verify that the sequence of data fields on the data record matches the listed sequence of attributes in the metafile.
- Verify that the actual data fields are delimited exactly as specified in the delimiter specification of the ATTRIBUTES statement.
- For FILE Data Providers, verify that only one file source (SOURCE FILE statement) is specified for each attribute group (NAME statement) or that you have used ManagedSystemName to distinguish the sources.
- For SOCK Data Providers, verify that you have the correct socket source host name (SOURCE SOCK) specified for the application.
- Examine the UAGENT DPLOG report in Tivoli Enterprise Portal. It might include messages that help the solution of the problems.

IBM Tivoli Universal Agent does not start

The common reason for the Universal Agent failing to start up is that the Universal Agent could not allocate the DCH port 1919. Example 9-14 shows the error log indicating that UA could not be started.

Example 9-14 RAS1 log

```
kum0sock.c,110,"KUMO_OpenLocalSocket") bind failed for local address UDP socket
512, port=1919, = error=10048 kum0sock.c,110, "KUMO_OpenLocalSocket") bind
failed for local address TCP socket 512, port=1919, error=10048
kumdsock.cpp,964, "ipcSock::allocateDCHport") Error: Could not open TCP/UDP
sockets bound to universal agent DCH port 1919
kumdsock.cpp,965,"ipcSock::allocateDCHport") Determine if another copy of
Universal Agent is already active on this system. Exiting...
```

This occurs when another Universal Agent is running in the same system, or there is another process allocating the port 1919 in the system. If there is another process allocating the same port, we can change the startup port for the Universal Agent using the KUMA_DCH_PORT environment variable in the KUMENV file in Windows systems or the um.ini file in UNIX systems, and then define a new port for UA startup.

IBM Tivoli Universal Agent–emitted traps

To prevent the possibility of different policy-generated traps being sent to a third-party SNMP Manager product with the same severity, set the following Universal Agent environment variable:

```
KUMP_TRAP_USE_POLICY_SEVERITY=Y
```

Managed System version suffix increasing

The Managed System version suffix is incremented when the Universal Agent is restarted even though the metafile has not changed. This can happen if there are two metafiles with the same three-character application prefix being activated at the same time. Universal Agent determines that the three-character application has been modified, then increments the version suffix.

Turning on additional tracing

Table 9-10 lists environment variables that can be used to turn on additional tracing over Universal Agent and its data providers.

Table 9-10 Environment variable for tracing

Variable	Used for
KDC_DEBUG=Y	Diagnosing communications and connectivity problems between Universal Agent and the Tivoli Enterprise Portal Server.
KUMA_VERBOSE=Y	Tracing a Universal Agent API client program. This variable must be set on the machine where the API client program is executing, not where Universal Agent is running. For example: `KUMP_API_VERBOSE=Y ^>dpapi.log` The KUMP_API_VERBOSE option is valuable when debugging an API program that communicates with the Universal Agent API data provider.
KBB_RAS1=ERROR (UNIT:kumaeagt ALL) (UNIT:kumpemit ALL) KUMA_DCH_TRAPEMIT=Y	SNMP Emitter tracing. Use to display emitted traps in the UAGENT Action report.

Variable	Used for
KUMP_ODBC_DEBUG=Y	ODBC Data Provider tracing.
KUMP_HTTP_DEBUG=Y	HTTP Data Provider tracing.
KUMP_SCRIPT_DEBUG=Y	Script Data Provider tracing.
KUMP_SNMP_DEBUG_TRAP=Y KUMP_SNMP_DEBUG_DISCOVERY_ROUTE=Y KUMP_SNMP_DEBUG_DISCOVERY_NETWORK=Y KUMP_SNMP_DEBUG_MIB_MANAGER=Y KUMP_SNMP_DEBUG_MIB_IO=Y	SNMP Data Provider tracing. All of the debug environment variables listed above default to No. As an example, if you use the SNMP Data Provider and have problems collecting MIB data, you set these two environment variables: KUMP_SNMP_DEBUG_MIB_MANAGER=Y KUMP_SNMP_DEBUG_MIB_IO=Y
ERROR (UNIT:kumpfile Error State Detail Flow Metrics) (UNIT:kumpdcmf ALL)	Detailed File Data Provider tracing.
ERROR (UNIT:kumpsosr ALL) (UNIT:kumpspst ALL) (UNIT:kumpscku ALL) (UNIT:kumpstcp ALL) (UNIT:kumplpba ALL)	Detailed API or Socket Data Provider tracing.
ERROR (UNIT:kumamain ALL)	Problems involving managed system online/offline processing.
ERROR (UNIT:kumpdpda Error Output) (UNIT:kumpmd2a Error Detail)	Incorrect report data.
ERROR METRICS	Problems involving Universal Agent memory usage.

10

Integration with IBM Tivoli Enterprise Console

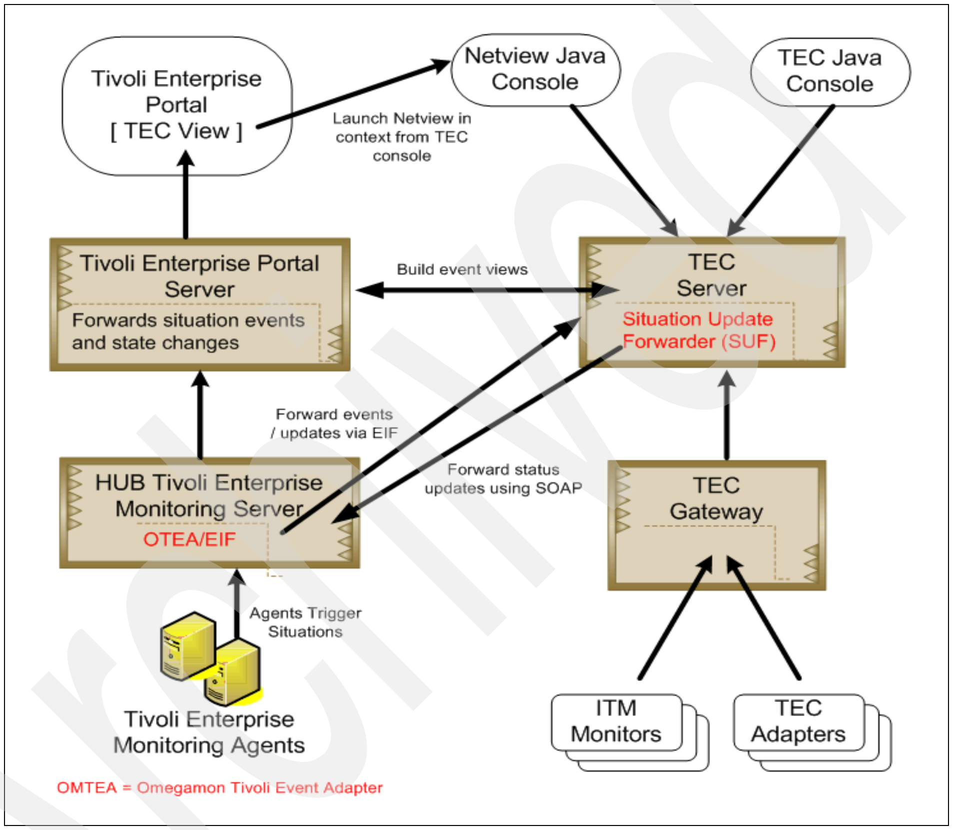
IBM Tivoli Monitoring 6.1 can be configured to send events to IBM Tivoli Enterprise Console, the IBM strategic event-management program. This chapter describes how to integrate IBM Tivoli Monitoring 6.1 with the IBM Tivoli Enterprise Console.

This chapter has the following sections:

- IBM Tivoli Monitoring 6.1 and IBM Tivoli Enterprise Console integration overview
- OMEGAMON IBM Tivoli Enterprise Console Event Adapter (OTEA)
- Event and rule definitions
- Situation Update Forwarder
- Synchronizing IBM Tivoli Monitoring situations and Tivoli Enterprise Console events

10.1 IBM Tivoli Monitoring 6.1 and IBM Tivoli Enterprise Console integration overview

Figure 10-1 shows an overview of the integration between IBM Tivoli Monitoring 6.1 and IBM Tivoli Enterprise Console.



Tivoli Enterprise Portal [TEC View]
Netview Java Console
TEC Java Console
Launch Netview in context from TEC console
Tivoli Enterprise Portal Server
Forwards situation events and state changes
Build event views
TEC Server
Situation Update Forwarder (SUF)
Forward events / updates via EIF
Forward status updates using SOAP
HUB Tivoli Enterprise Monitoring Server
OTEA/EIF
TEC Gateway
Agents Trigger Situations
Tivoli Enterprise Monitoring Agents
ITM Monitors
TEC Adapters
OMTEA = Omegamon Tivoli Event Adapter

Figure 10-1 IBM Tivoli Monitoring 6.1 and TEC Integration summary

In this integration, IBM Tivoli Enterprise Console is the Event Management focal point for all enterprise events, coming from sources such as IBM Tivoli Storage Manager, Networks, IBM Tivoli Configuration Manager, and third-party products (for example, Compaq Insight Manager). The standard interfaces for event synchronization are delivered out of the box, with predefined rules for handling various event update and synchronization scenarios (described in 10.4, “Event and rule definitions” on page 600).

The various components used in this integration are:

- OMEGAMON Tivoli Event Adapter (OTEA)

  This is supplied as part of the Tivoli Enterprise Monitoring Server (TEMS), and EIF is the communication method used to send the IBM Tivoli Enterprise Console events.

- TEC event synchronization

  This component is installed on each IBM Tivoli Enterprise Console server that should receive events from IBM Tivoli Monitoring 6.1. The underlying component is called the Situation Update Forwarder (SUF), and SOAP is used as the communication method to send updates to the TEMS.

- Event and rule definitions
  - omegamon.rls: This contains event-specific synchronization rules.
  - omegamon.baroc: This contains the new IBM Tivoli Monitoring 6.1 base event definitions.
  - Sentry.baroc: This contains an updated version of Sentry2_0_Base to support Distributed Monitoring 3.x environments.
  - k<TAG>.baroc: Each IBM Tivoli Monitoring 6.1 monitoring component will provide separate files containing specific monitoring components. The <TAG> will be 3z, a4, ex, ib, nt, oq, or, oy, ud, ul, ux. For example, the Windows events will be defined in knt.baroc.

    These files must be imported into your rulebase manually after the installation.

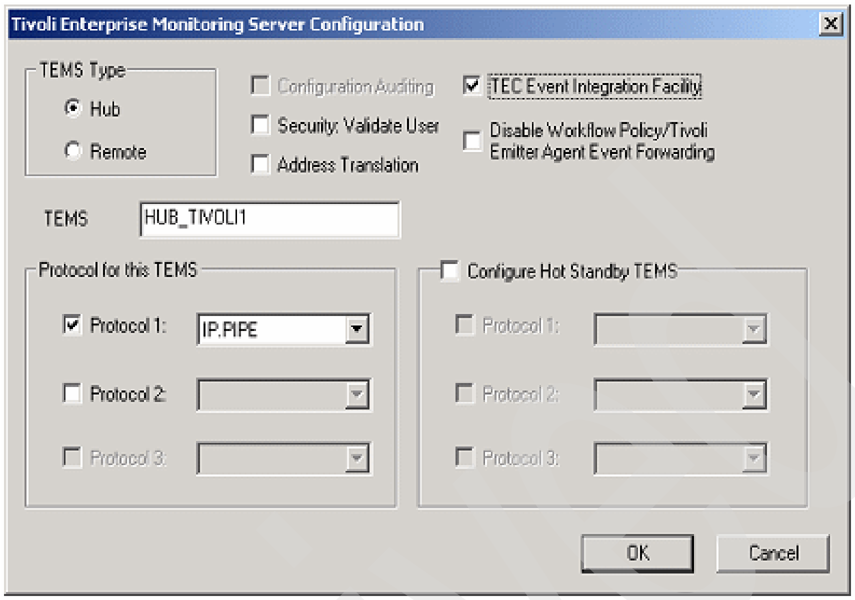
10.2 OMEGAMON IBM Tivoli Enterprise Console Event Adapter (OTEA)

This section discusses installing and configuring OMEGAMON IBM Tivoli Enterprise Console Event Adapter (OTEA).

10.2.1 Installing the OTEA

To install the OTEA:

1. Enable the **TEC Event Integration Facility** option during the installation of the Tivoli Enterprise Monitoring Server (TEMS), as shown in Figure 10-2 on page 586.


Tivoli Enterprise Monitoring Server Configuration

TEMS Type: [✓ Hub] Remote

- [ ] Configuration Auditing
- [ ] Security: Validate User
- [ ] Address Translation
- [x] TEC Event Integration Facility
- [ ] Disable Workflow Policy/Tivoli Emitter Agent Event Forwarding

TEMS: HUB_TIVOLI1

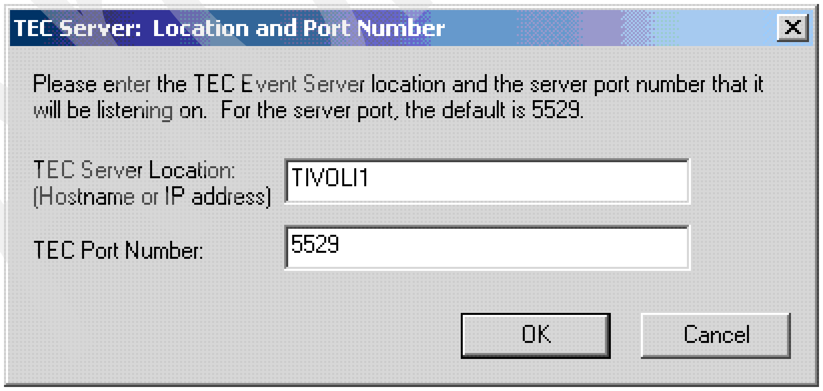
Protocol for this TEMS
- [x] Protocol 1: IP.PIPE
- [ ] Protocol 2:
- [ ] Protocol 3:

- [ ] Configure Hot Standby TEMS

Figure 10-2   Installing OMEGAMON TEC Adapter (OTEA)

2. When you click **OK**, you will configure the HUB TEMS. Then you are prompted to enter the host name and port number of your IBM Tivoli Enterprise Console server to which events will be forwarded (Figure 10-3).

   Note: Event forwarding uses the non-TME based transport mechanism.


TEC Server: Location and Port Number

Please enter the TEC Event Server location and the server port number that it will be listening on. For the server port, the default is 5529.

TEC Server Location: (Hostname or IP address): TIVOLI1

TEC Port Number: 5529

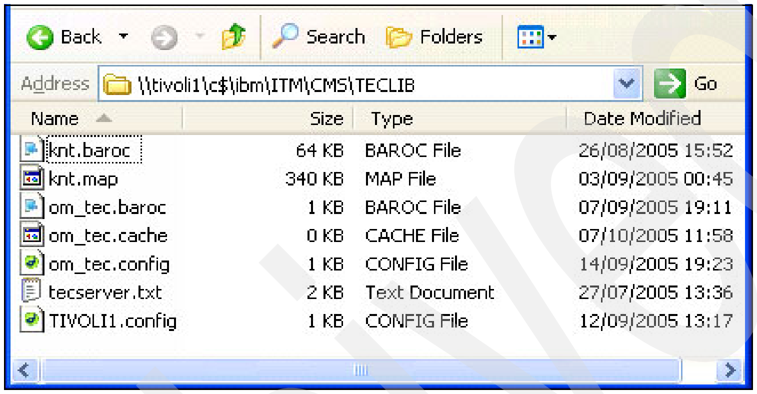
Figure 10-3   Enter target TEC server host name and port number

10.2.2 Configuring the OTEA

The configuration files for the OTEA are in different locations, depending on the platform type of your TEMS server:

Windows	<itm install dir>\tems\TECLIB
UNIX	<itm install dir>/ tables/<tems hostname>/TECLIB

Figure 10-4 shows the files contained in this location.



Name	Size	Type	Date Modified
knt.baroc	64 KB	BAROC File	26/08/2005 15:52
knt.map	340 KB	MAP File	03/09/2005 00:45
om_tec.baroc	1 KB	BAROC File	07/09/2005 19:11
om_tec.cache	0 KB	CACHE File	07/10/2005 11:58
om_tec.config	1 KB	CONFIG File	14/09/2005 19:23
tecserver.txt	2 KB	Text Document	27/07/2005 13:36
TIVOLI1.config	1 KB	CONFIG File	12/09/2005 13:17

Figure 10-4 OMEGAMON TEC Adapter configuration files

EIF configuration

om_tec.config is the EIF configuration file. It contains several Tivoli Enterprise Console Adapter keywords, as shown in Example 10-1.

Example 10-1 om_tec.config file

```
ServerLocation=TIVOLI1
ServerPort=5529
EventMaxSize=4096
NO_UTF8_CONVERSION=YES
ConnectionMode=co
BufferEvents=YES
BufEvtMaxSize=4096
BufEvtPath=./TECLIB/om_tec.cache
FilterMode=OUT
Filter:Class=ITM_Generic;master_reset_flag=''
```

Extra filters can be added to this file to filter specified events.

The FilterMode and Filter keywords

Depending on how you specify the Filter and FilterMode keywords, filtered events are either sent to the event server or discarded. To send specific events to the event server:

1. Set FilterMode to IN.
2. Create Filter statements to match the specific events that you want sent.

To discard specific events:

1. Set FilterMode to OUT (the default value).
2. Create Filter statements to match the specific events that you want to discard.

Example 10-2 shows two filters. The first filter suppresses all events of type ITM_Generic, and the second suppresses all events of type ITM_DNS_Query from the IP address 192.168.1.101.

Example 10-2 Example filtering

```
FilterMode=OUT
Filter:Class=ITM_Generic;
Filter:Class=ITM_DNS_Query;origin=192.168.1.101
```

A detailed explanation of Event Adapter filtering can be found in *IBM Tivoli Event Integration Facility Reference, Version 3.9*, SC32-1241.

Event severity

The *tec_server.txt* file contains the non-default event severity mapping. By default, the IBM Tivoli Monitoring 6.1 severity of Informational is mapped to a Tivoli Enterprise Console severity of HARMLESS, and the IBM Tivoli Monitoring severities of WARNING and CRITICAL are preserved in the Tivoli Enterprise Console severity. The resulting Tivoli Enterprise Console severities of specific IBM Tivoli Monitoring situations can be controlled by modifying this tec_server.txt file.

In this file lines starting with an asterisk (*) in column one are treated as comments. Each line specifies a destination IBM Tivoli Enterprise Console server and optionally a Tivoli Enterprise Console severity level for a situation name.

The syntax of each line is as follows:*

```
sitname=tecservername[:port]|*[,SEVERITY=severitylevel]
```

In this syntax:

- `tecservername` is the target IBM Tivoli Enterprise Console server name, address or '*' that denotes the default IBM Tivoli Enterprise Console server.

- `port` is the port the IBM Tivoli Enterprise Console server is listening on, or 0 if the IBM Tivoli Enterprise Console server is using port mapping (UNIX serves). If no port is specified, the default will be 0 (use port mapping).
- `severitylevel` is one of the valid severity levels supported by the IBM Tivoli Enterprise Console server (FATAL | CRITICAL | MINOR | WARNING | HARMLESS | UNKNOWN).

For example, the two entries in Example 10-3 cause IBM Tivoli Enterprise Console events with CRITICAL and WARNING severity be sent to the IBM Tivoli Enterprise Console server tec1 and the default IBM Tivoli Enterprise Console server, respectively, when the situations Sit1 and Sit2 are raised.

Example 10-3 sitname example

```
Sit1=tec1,SEVERITY=CRITICAL
Sit2=*,SEVERITY=WARNING
```

The clearing events for IBM Tivoli Monitoring 6.1 situation events are always sent with the same severity as the original event.

If either om_tec.config or tec_server.txt is modified, the TEMS must be restarted. These files can be configured by right-clicking the **Tivoli Enterprise Monitoring Server** entry in the Manage Tivoli Enterprise Monitoring Services application, and selecting **Advanced**, as shown in Figure 10-5.

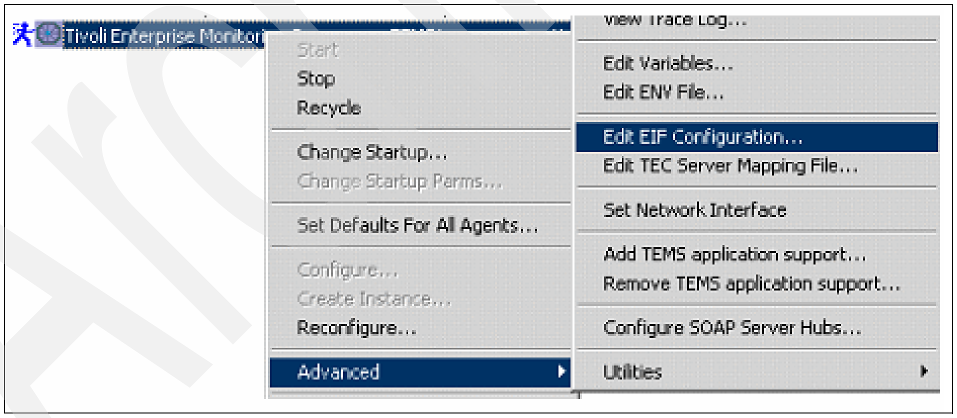

Tivoli Enterprise Monitor
Start
Stop
Recycle
Change Startup...
Change Startup Parms...
Set Defaults For All Agents...
Configure...
Create Instance...
Reconfigure...
Advanced
View Trace Log...
Edit Variables...
Edit ENV File...
Edit EIF Configuration...
Edit TEC Server Mapping File...
Set Network Interface
Add TEMS application support...
Remove TEMS application support...
Configure SOAP Server Hubs...
Utilities

Figure 10-5 Configuring the OMEGAMON IBM Tivoli Enterprise Console Adapter

In our environment, when this method was used to open the Tivoli Enterprise Console server Mapping file (tec_server.txt), the file was launched in Notepad with UNIX-style linefeeds. This made it very difficult to modify the file. Hence the file was modified directly, using vi.

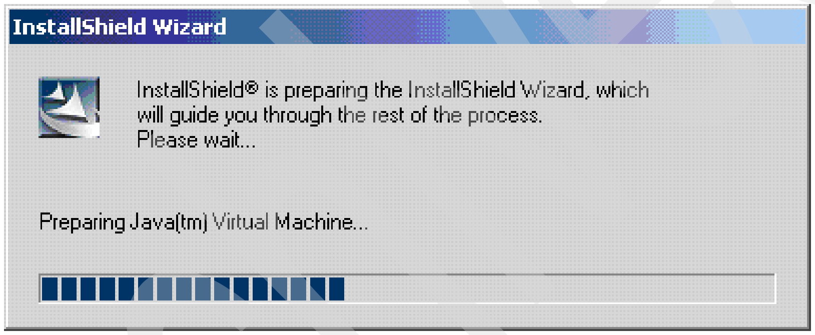
10.3 Installing TEC event synchronization

The TEC event synchronization component must be installed on the IBM Tivoli Enterprise Console server. In our test environment, a Windows IBM Tivoli Enterprise Console server was used, with IBM Tivoli Enterprise Console 3.9 and Fix Pack 03 installed.

Note: To install TEC event synchronization on a UNIX IBM Tivoli Enterprise Console, refer to 3.2.14, “Event synchronization installation” on page 147.

The installation program for the TEC event synchronization component can be found on the IBM Tivoli Monitoring 6.1 installation media in the TEC folder. Follow these steps:

1. Double-click **setupwin32.exe** to initialize the installation wizard (Figure 10-6).

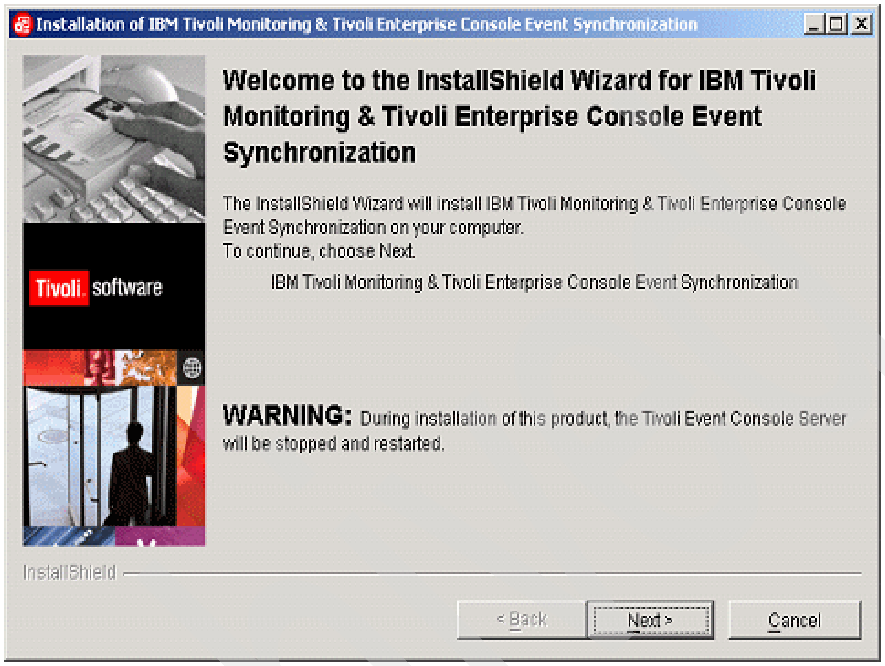


InstallShield® is preparing the InstallShield Wizard, which will guide you through the rest of the process.
Please wait...

Preparing Java(tm) Virtual Machine...

Figure 10-6 Installation wizard initialization

2. Be aware that during the installation, the IBM Tivoli Enterprise Console server will be restarted (Figure 10-7). Click **Next** to continue.


Installation of IBM Tivoli Monitoring & Tivoli Enterprise Console Event Synchronization

Welcome to the InstallShield Wizard for IBM Tivoli Monitoring & Tivoli Enterprise Console Event Synchronization

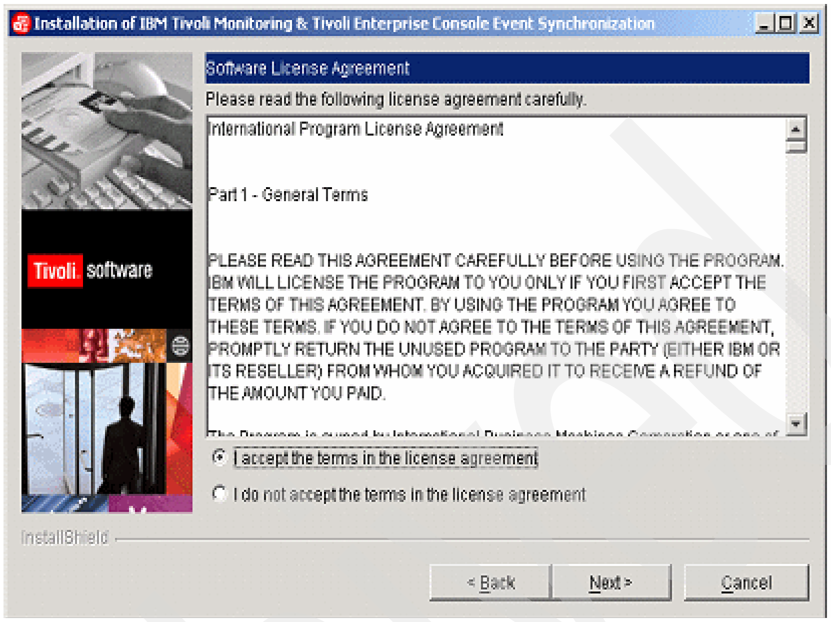
The InstallShield Wizard will install IBM Tivoli Monitoring & Tivoli Enterprise Console Event Synchronization on your computer.
To continue, choose Next.

IBM Tivoli Monitoring & Tivoli Enterprise Console Event Synchronization

WARNING: During installation of this product, the Tivoli Event Console Server will be stopped and restarted.

Figure 10-7 Installation welcome window

3. Accept the license agreement and click **Next** (Figure 10-8).


Installation of IBM Tivoli Monitoring & Tivoli Enterprise Console Event Synchronization
Tivoli software
InstallShield
< Back
Next >
Cancel

Software License Agreement

Please read the following license agreement carefully.

International Program License Agreement

Part 1 - General Terms

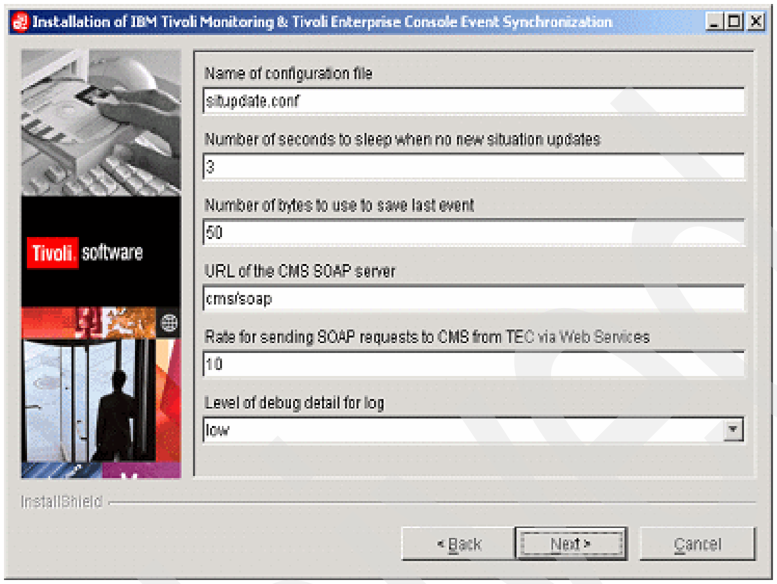
PLEASE READ THIS AGREEMENT CAREFULLY BEFORE USING THE PROGRAM. IBM WILL LICENSE THE PROGRAM TO YOU ONLY IF YOU FIRST ACCEPT THE TERMS OF THIS AGREEMENT. BY USING THE PROGRAM YOU AGREE TO THESE TERMS. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT, PROMPTLY RETURN THE UNUSED PROGRAM TO THE PARTY (EITHER IBM OR ITS RESELLER) FROM WHOM YOU ACQUIRED IT TO RECEIVE A REFUND OF THE AMOUNT YOU PAID.

[✓ I accept the terms in the license agreement]

I do not accept the terms in the license agreement

Figure 10-8 Accept the license agreement

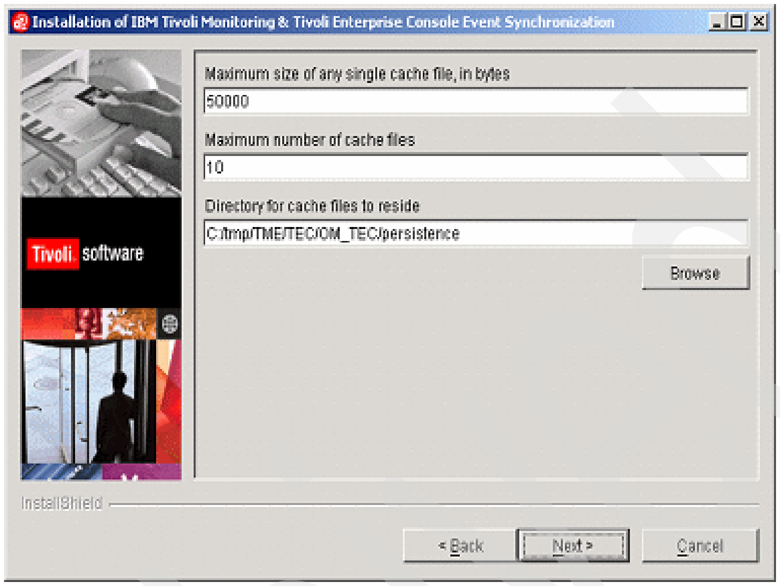
4. Configuration values can be customized now or later (Figure 10-9). Click **Next** to continue.


Installation of IBM Tivoli Monitoring & Tivoli Enterprise Console Event Synchronization
Tivoli. software
InstallShield
< Back
Next >
Cancel

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 CMS SOAP server
cms/soap
Rate for sending SOAP requests to CMS from TEC via Web Services
10
Level of debug detail for log
low

Figure 10-9 Customizing configuration values

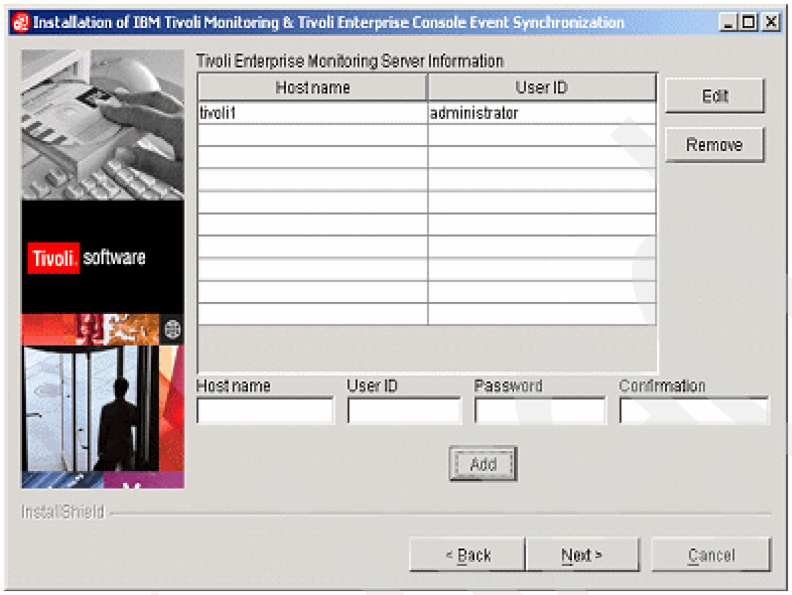
5. Customize the maximum size of the cache file, and click **Next** to continue (Figure 10-10).


Installation of IBM Tivoli Monitoring & Tivoli Enterprise Console Event Synchronization
Tivoli. software
InstallShield
Browse
< Back
Next >
Cancel

Maximum size of any single cache file, in bytes
50000
Maximum number of cache files
10
Directory for cache files to reside
C:/tmp/TME/TEC/OM_TEC/persistence

Figure 10-10 Set the maximum size of the cache file

6. Enter details of your TEMS and user credentials. Click **Next** (Figure 10-11).


Installation of IBM Tivoli Monitoring & Tivoli Enterprise Console Event Synchronization

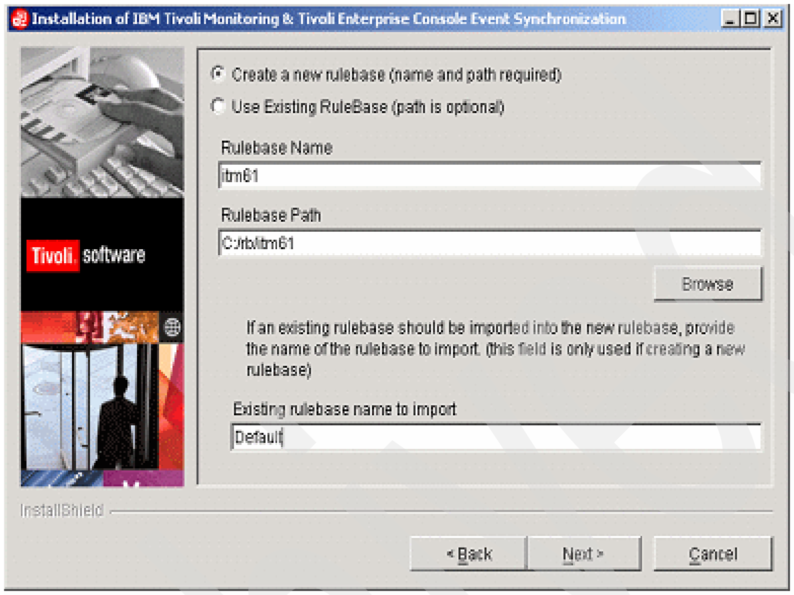
Tivoli Enterprise Monitoring Server Information

Host name	User ID
tivoli1	administrator

Host name | User ID | Password | Confirmation

Figure 10-11 Details of your TEMS

7. You can now create a new rulebase or use an existing rulebase. Also, existing event definitions and rule sets can be imported from another rulebase (Figure 10-12). Click **Next**.



Installation of IBM Tivoli Monitoring & Tivoli Enterprise Console Event Synchronization

(● Create a new rulebase (name and path required)) ○ Use Existing RuleBase (path is optional)

Rulebase Name: itm61

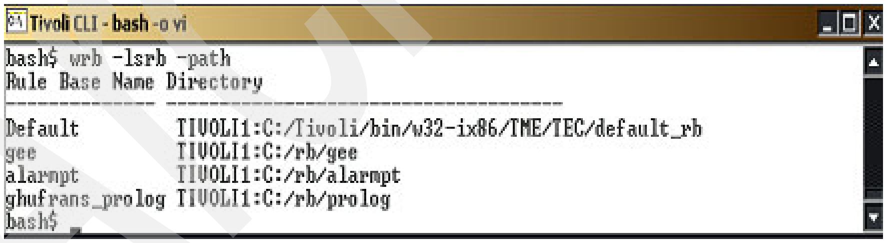
Rulebase Path: C:/rb/itm61

If an existing rulebase should be imported into the new rulebase, provide the name of the rulebase to import. (this field is only used if creating a new rulebase)

Existing rulebase name to import: Default

Figure 10-12 Create a new rulebase or use an existing rulebase window

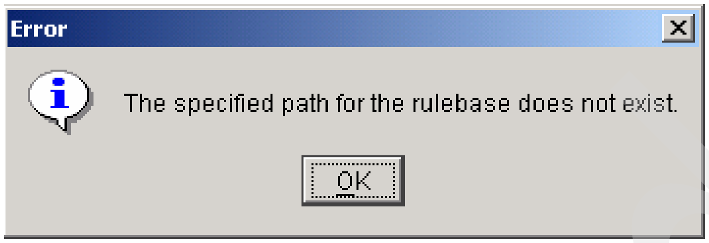
8. From a command prompt, you can verify the correct location of existing rulebases before creating a new one, as shown in Figure 10-13.



```
bash$ wrb -lsrb -path
Rule Base Name Directory
-------------- -------------------------------------
Default        TIVOLI1:C:/Tivoli/bin/w32-ix86/TME/TEC/default_rb
gee            TIVOLI1:C:/rb/gee
alarmpt        TIVOLI1:C:/rb/alarmpt
ghufrans_prolog TIVOLI1:C:/rb/prolog
bash$ _
```

Figure 10-13 Verify the correct location of existing rulebases

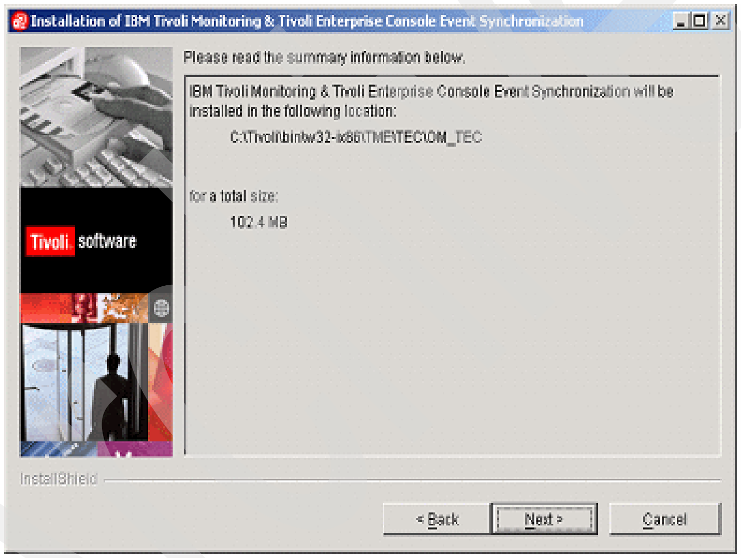
Note that the new rulebase directory must exist, otherwise an error will be displayed as shown in Figure 10-14.


Error

The specified path for the rulebase does not exist.

Figure 10-14 The new rulebase directory must exist

9. Click **Next** to continue the installation (Figure 10-15).


Installation of IBM Tivoli Monitoring & Tivoli Enterprise Console Event Synchronization

Please read the summary information below.

IBM Tivoli Monitoring & Tivoli Enterprise Console Event Synchronization will be installed in the following location:

C:\Tivoli\bin\w32-ix86\TME\TEC\OM_TEC

for a total size:

102.4 MB

Figure 10-15 Summary information

10. The installation continues, the rulebase is configured, and the IBM Tivoli Enterprise Console server is restarted (Figure 10-16).

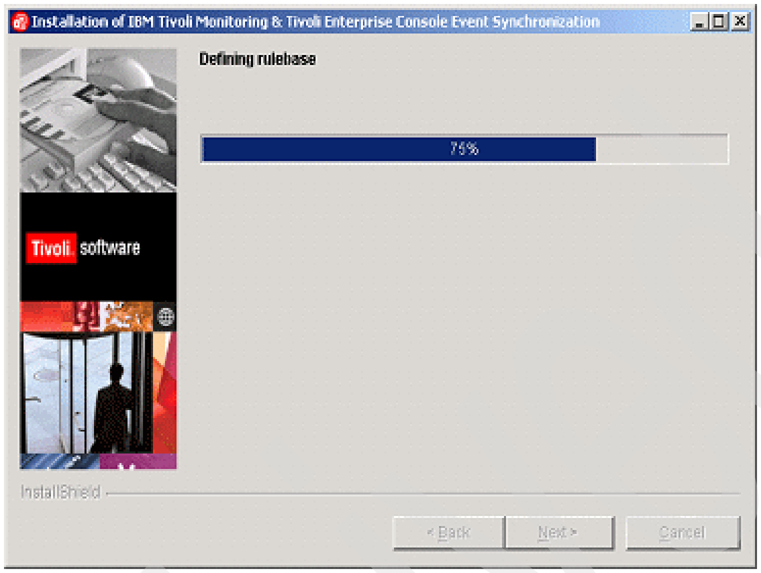
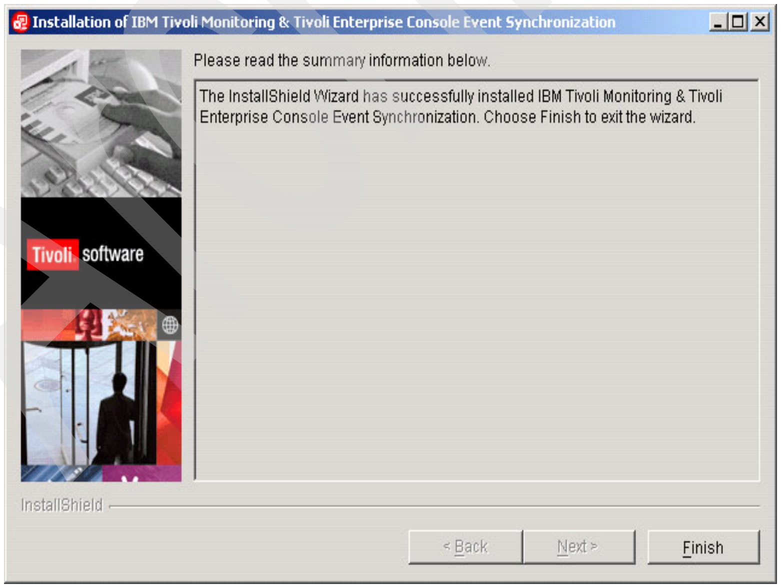

Installation of IBM Tivoli Monitoring & Tivoli Enterprise Console Event Synchronization
Defining rulebase
75%
Tivoli software
InstallShield
< Back
Next >
Cancel

Figure 10-16 Rulebase definition

11. The installation completes successfully, as shown in Figure 10-17.


Installation of IBM Tivoli Monitoring & Tivoli Enterprise Console Event Synchronization
Tivoli software
InstallShield
< Back
Next >
Finish

Please read the summary information below.

The InstallShield Wizard has successfully installed IBM Tivoli Monitoring & Tivoli Enterprise Console Event Synchronization. Choose Finish to exit the wizard.

Figure 10-17 Success notice

You can verify that the rulebase directory has been created successfully, as shown in Figure 10-18.

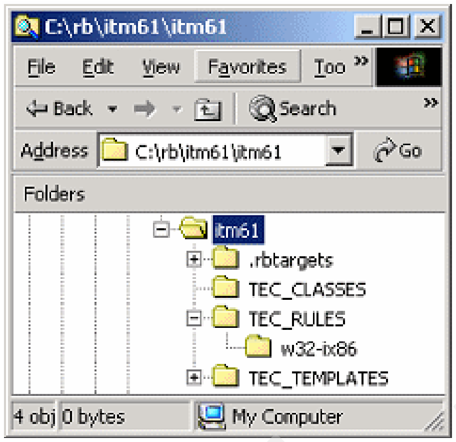
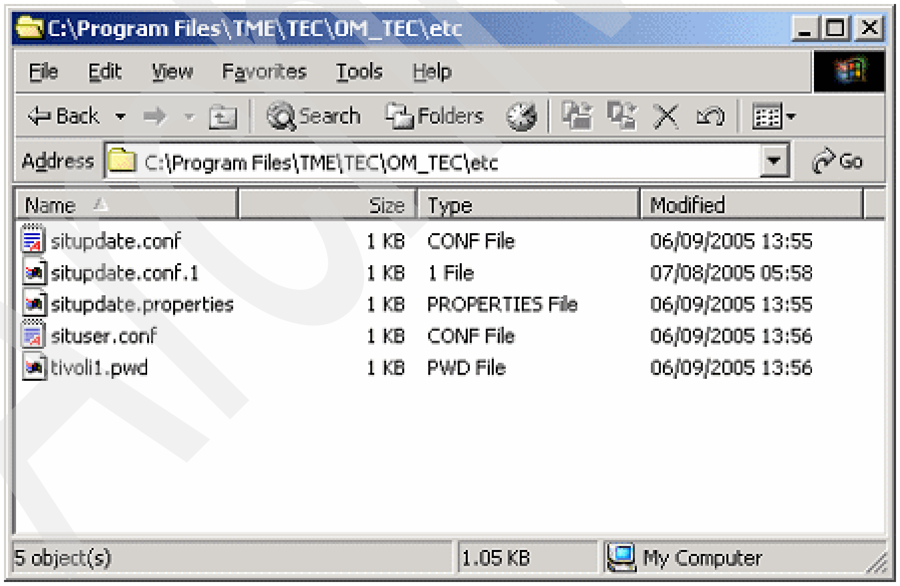


Figure 10-18 Verify that the rulebase directory has been created

Figure 10-19 shows the default location for the synchronization configuration files.


C:\Program Files\TME\TEC\OM_TEC\etc
File Edit View Favorites Tools Help
Back Search Folders
Address C:\Program Files\TME\TEC\OM_TEC\etc Go

Name	Size	Type	Modified
situpdate.conf	1 KB	CONF File	06/09/2005 13:55
situpdate.conf.1	1 KB	1 File	07/08/2005 05:58
situpdate.properties	1 KB	PROPERTIES File	06/09/2005 13:55
situser.conf	1 KB	CONF File	06/09/2005 13:56
tivoli1.pwd	1 KB	PWD File	06/09/2005 13:56

5 object(s) 1.05 KB My Computer

Figure 10-19 Default location for the synchronization configuration files

The synchronization log file is located in C:\tmp\itmsynch\logs (Figure 10-20).

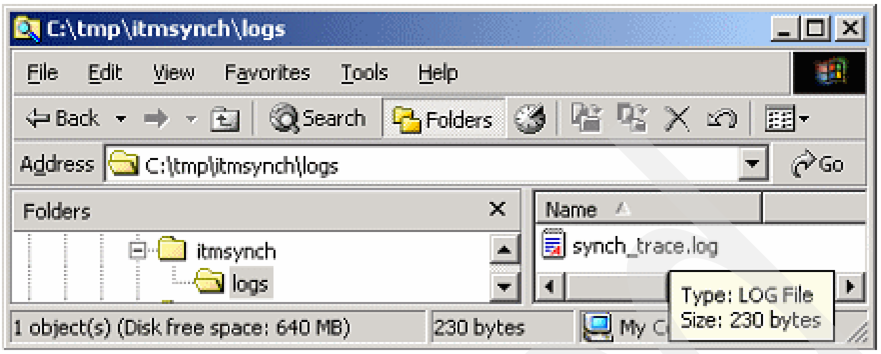


Figure 10-20 The synchronization log file is located in C:\tmp\itmsynch\logs

10.4 Event and rule definitions

In this section we discuss event and rule definitions.

10.4.1 Event definitions

The file omegamon.baroc contains the following event definitions:

- Omegamon_Base

  This event inherits from EVENT and has the following specific attribute definitions:

source	Defaults to `OMEGAMON`.
sub_source	Defaults to `OM_TEC`.
cms_hostname	Contains the host name of the managed system where the event originates.
cms_port	Contains the port number that the Tivoli Enterprise Monitoring Server Web service listens on.
integration_type	Indicates whether the event is a new (`N`) or update (`U`) event.
master_reset_flag	This is set when the event is a “master reset” event; set to `R` when the TEMS is restarted or `S` when the hot standby is restarted. Otherwise, the attribute is set to `NULL`.
appl_label	Contains any application-specific data.
situation_name	Contains the name of the situation.

situation_origin	Contains the managed system name where the event originated.
situation_displayitem	Contains the display item of the situation if available.
situation_time	Contains the 16-byte timestamp of the situation event.
situation_status	Contains the current status of the situation event.
situation_eventdata	Contains the event data attributes in key-pair format.

- Omegamon_Generic

  This event inherits from Omegamon_Base and has no specific attribute definitions.

- TEC_ITM_OM_Situation_Sync_Error

  This event inherits from Omegamon_Base and has no specific attribute definitions.

- source:
- TEC_ITM_Config_Sync

  This event inherits from EVENT and has the following specific attribute definitions:

fileSize	INTEGER, default=50000
fileNumber	INTEGER, default=10
fileLocation	STRING
pollingInterval	INTEGER, default=3
crcByteCount	INTEGER
cmsSoapUrl	STRING, default="cms/soap"
bufferFlushRate	INTEGER
logLevel	STRING, default="low"

The file Sentry.baroc contains a new definition for the Sentry2_0_Base event, which now has the following extra attributes:

cms_hostname	Contains the host name of the managed system where the event originates.
cms_port	Contains the port number that the Tivoli Enterprise Monitoring Server Web service is listening on.
integration_type	Indicates whether the event is a new (`N`) or update (`U`) event.
master_reset_flag	This is set when the event is a “master reset” event; set to `R` when the TEMS is restarted, or `S` when the

	hot standby is restarted. Otherwise the attribute is set to NULL.
appl_label	Contains any application-specific data.
situation_name	Contains the name of the situation.
situation_origin	Contains the managed system name where the event originated.
situation_displayitem	Contains the display item of the situation if available.
situation_time	Contains the 16-byte timestamp of the situation event.
situation_status	Contains the current status of the situation event.
situation_eventdata	Contains the event data attributes in key-pair format.

The file om_tec.baroc contains the event definitions for ITM_Generic, which inherits from Omegamon_Base, and has no specified attributes defined.

IBM Tivoli Monitoring 6.1 situations are mapped to corresponding event classes that the attribute group belongs to. Each IBM Tivoli Monitoring 6.1 component (agent) has its own base class, such as KNT_Base for Windows, shown in Example 10-4.

Example 10-4 Example IBM Tivoli Monitoring 6.1 event definition: ITM_NT_System

```
TEC_CLASS :
        ITM_NT_System ISA KNT_Base
        DEFINES {

                #
                # NT_System attribute group
                #

                server_name:STRING;
                timestamp:STRING;
                user_name:STRING;
                operating_system_type:STRING;
                operating_system_version:STRING;
                network_address:STRING;
                number_of_processors:INTEGER;
                processor_type:INTEGER;
                page_size:INTEGER;
                pct__total_privileged_time:INTEGER;
                pct__total_processor_time:INTEGER;
                pct__total_user_time:INTEGER;
                context_switches_per_sec:INTEGER;
                file_control_bytes_per_sec:INTEGER;
                file_control_operations_per_sec:INTEGER;
                file_data_operations_per_sec:INTEGER;
                file_read_bytes_per_sec:INTEGER;
                file_read_operations_per_sec:INTEGER;
                file_write_bytes_per_sec:INTEGER;
                file_write_operations_per_sec:INTEGER;
                processor_queue_length:INTEGER;
                system_calls_per_sec:INTEGER;
                system_up_time:INTEGER;
                total_interrupts_per_sec:INTEGER;
                alignment_fixups_per_sec:INTEGER;
                exception_dispatches_per_sec:INTEGER;
                floating_emulations_per_sec:INTEGER;
                user_name_u:STRING;

        };
END
```

Each attribute of the attribute group has its own attribute in the associated event class, as shown in Example 10-5.

Example 10-5 Example IBM Tivoli Monitoring 6.1 event definition: ITM_NT_Memory

```
TEC_CLASS :
        ITM_NT_Memory ISA KNT_Base
        DEFINES {
                server_name:STRING;
                timestamp:STRING;
                available_bytes:INTEGER;
                cache_bytes:INTEGER;
                cache_bytes_peak:INTEGER;
                cache_faults_per_sec:INTEGER;
                commit_limit:INTEGER;
                committed_bytes:INTEGER;
                demand_zero_faults_per_sec:INTEGER;
                free_system_page_table_entries:INTEGER;
                page_faults_per_sec:INTEGER;
                page_reads_per_sec:INTEGER;
                page_writes_per_sec:INTEGER;
                pages_input_per_sec:INTEGER;
                pages_output_per_sec:INTEGER;
                pages_per_sec:INTEGER;
                pool_paged_allocs:INTEGER;
                pool_paged_bytes:INTEGER;
                pool_nonpaged_allocs:INTEGER;
                pool_nonpaged_bytes:INTEGER;
                transition_faults_per_sec:INTEGER;
                write_copies_per_sec:INTEGER;
                available_kbytes:INTEGER;
                cache_kbytes:INTEGER;
                cache_kbytes_peak:INTEGER;
                commit_limit_kb:INTEGER;
                committed_kbytes:INTEGER;
                pool_paged_kbytes:INTEGER;
                pool_nonpaged_kbytes:INTEGER;
                pool_paged_resident_bytes:INTEGER;
                system_cache_resident_bytes:INTEGER;
                system_code_total_bytes:INTEGER;
                system_driver_resident_bytes:INTEGER;
                system_driver_total_bytes:INTEGER;
};
END
```

10.4.2 OMEGAMON rules

The file omegamon.rls contains a number of rules, which are grouped into three sections:

- Initialization section
- Synchronization section
- Update section

Initialization section

This section performs initialization and shutdown tasks, as well as the definition of some needed predicates to be used throughout the rule set.

- The ruleset behavior is defined in this section upon receiving a TEC_Start event. The following parameters are safe to be changed when reusing this rule:

omegamon_admin	The identifier used when a rule defined in this rule set closes an event. This identifier is used to differentiate close operations that were originated automatically rather than by the console operator.
om_config_file	This attribute specifies the name of the configuration file that will contain the values to be passed on to the short-running script situpdate.sh. This script will write CT_Acknowledge, CT_Reset, and CT_Resurface events to the event cache file, from which they will be sent to the Tivoli Enterprise Monitoring server/hub via Web services. The default name of the configuration file is situpdate.conf.
om_latency	This attribute specifies how far back go looking for events in the cache when trying to match events in the rules. The value is specified in seconds, and is used for both backward and forward event searches.

Note that the cache may or may not hold sufficient events that fall within the parameters of this attribute, so if you define a small event cache, there may be events that are still within this time period but will not be found by the search because the cache could not hold them and the event server has discarded them.

omsync_timeout This attribute sets the period in seconds that we must wait to distinguish between the synchronization of single or multiple events. Default timeout is 30 seconds.

omsync_maxentries This attribute sets the maximum number of hosts allowed per batch; one IBM Tivoli Enterprise Console task will be called per batch, so choose this number wisely because IBM Tivoli Enterprise Console tasks arguments are passed by command line and each operating system has a limit for the number of characters you may pass in the command line. Default batch size is 10 events.

sit_resurface_def_action

This attribute would be used to determine the default action of the rules if a situation update event arrives from Tivoli Enterprise Monitoring Server to resurface or reopen an event that has already been acknowledged. The two possible values are ACCEPT and REJECT, and the default value is ACCEPT.

sit_ack_expired_def_action

This attribute would be used to determine the default action of the rules if a situation update event arrives from Tivoli Enterprise Monitoring Server to reacknowledge an event that has already been acknowledged. This would happen when a situation's acknowledgement in Tivoli Enterprise Monitoring Server expires, and Tivoli Enterprise Monitoring Server reacknowledges the situation. The two possible values are ACCEPT and REJECT, and the default value is REJECT.

sf_check_timer This attribute specifies the interval at which the state of the long-running Java program would be checked. This daemon reads events from the cache files and sends them to the Tivoli Enterprise Monitoring Server via Web services. The default value is 600 seconds.

Default values of the configuration parameters required by the script situpdate.sh: fileSize is set to 50000, fileNumber is set to 10, and fileLocation is set to TME/TEC/OM_TEC/persistence.

omegamon_debug The debug flag; set to `yes` by default, it controls the output of this rule set to its own debug file. Make sure to set this flag to `no` flag in a production environment, because it can significantly affect the performance of this rule set.

omegamon_logfile The debug filename; the file where the debug messages are stored. If you do not specify an absolute location, this file will be created in the Tivoli database directory. The default filename is omegamon.log.

This section also contains a rule that is used to reread configuration parameters. It will be invoked if any configuration parameter changes dynamically while the IBM Tivoli Enterprise Console server is running, upon receiving an event of type TEC_ITM_Config_Sync.

Update section

This section contains rules that perform updates on events and are responsible for updating existing Omegamon_Base events with the information provided by a corresponding Omegamon_Base update event. If a corresponding event is not found in the event cache, then no update is performed. A corresponding base event is determined by matching the values of the situation_name, situation_host, and situation_displayitem fields.

The situation_displayitem may not be present in a base event. In this case only update events not containing a situation_displayitem will match. The type of update to be performed is based on the value of the situation_staus of the incoming update event and the current state of the corresponding base event. Table 10-1 shows the supported state transitions.

Table 10-1 Supported state transitions

Initial base state	Update event state	Final base state
Y	S	N
Y	P	P
any	Y	any
Y	N	N

Initial base state	Update event state	Final base state
Y	D	D
Y	A	A
A	E	Y
A	F	Y

This section also has a rule that handles the case wherein a Tivoli Enterprise Monitoring Server may be restarted or a hot standby switch takes place between two Tivoli Enterprise Monitoring Servers. In either case, all situation events from the Tivoli Enterprise Monitoring Server must be marked as having been reset because the Tivoli Enterprise Monitoring Server does not persist situations when the server goes down.

Synchronization section

This section is responsible for synchronizing status changes of events made in IBM Tivoli Enterprise Console with OMEGAMON. When a network element status event is acknowledged, closed, or leaves a previous acknowledged state (un-acknowledge), a synchronization trap is sent to OMEGAMON, so the OMEGAMON console can update the status associated with the network element immediately.

Note: This feature can be implemented only for interface, node, and router status events.

There is also a rule that synchronizes changes in interface events and sends traps so OMEGAMON can also take synchronization actions. An update to a synchronization event would trigger a change rule. This update can come from three sources:

- Tivoli Enterprise Console
- An IBM Tivoli Enterprise Console rule
- A Tivoli Enterprise Monitoring Server update event

IBM Tivoli Enterprise Console will send a CT_Acknowledge event to Tivoli Enterprise Monitoring Server only in the first two cases. This will be done by the rule using the admin event slot, which will be set to the name of the rule to indicate that a synchronization event should be sent to Tivoli Enterprise Monitoring Server.

Updated events are buffered so they can be synchronized with OMEGAMON. The op code is set based on the type of updated performed:

a	Acknowledge
c	Close
r	Resurface/reopen

The first entry added to an empty buffer starts a timer that flushes any buffered events when the timer expires (30 seconds). If the buffer contains 10 updates, then the buffer is flushed. Both the timer interval and the maximum number of events in the buffer are configurable.

There is also a rule to synchronize situation events over a Tivoli Enterprise Monitoring Server restart. After a server is restarted, the same situations may still exist on the agents and could be sent to Tivoli Enterprise Console as new situations. The Tivoli Enterprise Console would have a task: Resync Events. When the Tivoli Enterprise Console operator runs this task, all new situations or IBM Tivoli Monitoring events that have arrived from the Tivoli Enterprise Monitoring Server after it restarted will be compared with old events from the same server that have the master_reset_flag set to R. The slots being used for the comparison are situation_name, situation_origin, and situation_displayitem. If any matches are found, the older event will be linked to the newer event and then closed. The new event will be considered to represent the current state of the situation. This is not to say that there would be a 100% match, the reason being that information in the situation_displayitem for the same situation_name from the same situation_origin could change over time. However, this rule would link all IBM Tivoli Monitoring events for all situations that have not changed over a Tivoli Enterprise Monitoring Server recycle.

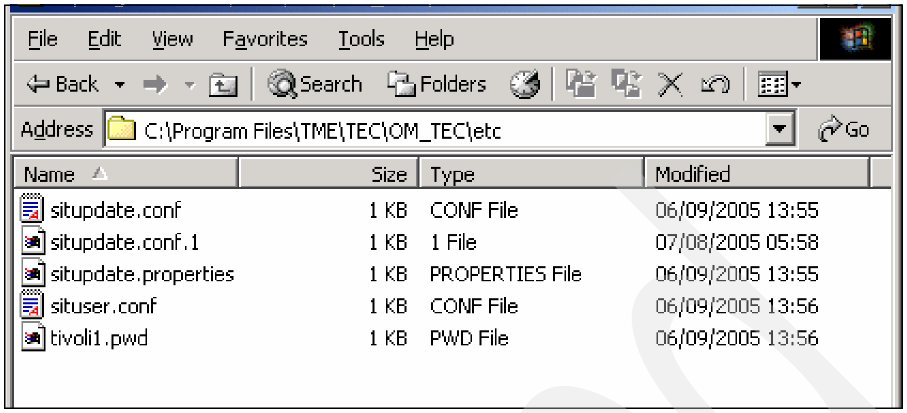
10.5 Situation Update Forwarder

The Situation Update Forwarder is the Web service based Java process that is used to communicate to the TEMS from IBM Tivoli Enterprise Console event server.

The configuration files for the Situation Update Forwarder are in different locations, depending on the platform type of your IBM Tivoli Enterprise Console server:

Windows	C:/Program Files/TEC/OM_TEC/etc
UNIX	$BINDIR/TME/TEC/OM_TEC/etc

Figure 10-21 on page 609 shows the files contained in this location.



Address: C:\Program Files\TME\TEC\OM_TEC\etc

Name	Size	Type	Modified
situpdate.conf	1 KB	CONF File	06/09/2005 13:55
situpdate.conf.1	1 KB	1 File	07/08/2005 05:58
situpdate.properties	1 KB	PROPERTIES File	06/09/2005 13:55
situser.conf	1 KB	CONF File	06/09/2005 13:56
tivoli1.pwd	1 KB	PWD File	06/09/2005 13:56

Figure 10-21 Contents of C:\Program File\TME\TEC\OM_TEC

situpdate.conf contains information about the configuration of the event synchronization, as shown below.

Example 10-6 Contents of situpdate.conf

```
fileSize=50000
fileNumber=10
fileLocation="C:/tmp/TME/TEC/OM_TEC/persistence"
pollingInterval=3
crcBytecount=50
cmsSoapUrl=cms/soap
bufferFlushRate=10
logLevel=low
```

situser.conf contains information about the user ID and password for interaction with the SOAP Web services, as shown below.

Example 10-7 Contents of situser.conf

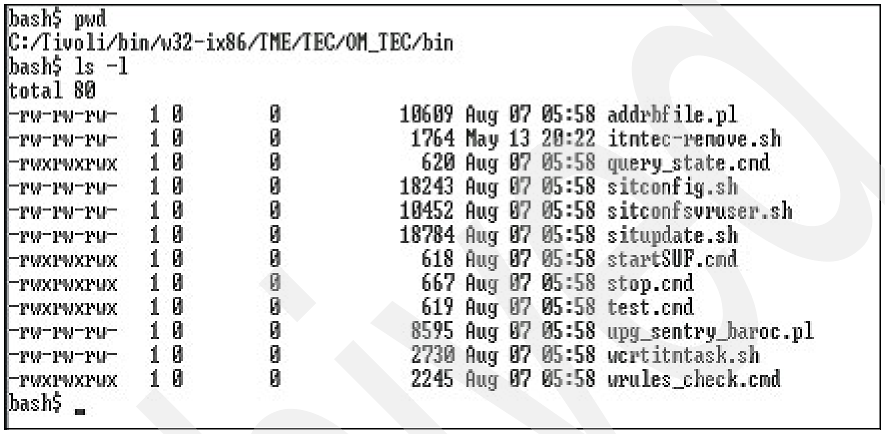
```
serverid=tivoli1
userid=administrator
passwordfile=c:/Program Files/TME/TEC/OM_TEC/etc/tivoli1.pwd
```

These files must only be modified with the scripts provided with the Situation Update Forwarder, and any change requires the Situation Update Forwarder to be restarted.

The Situation Update Forwarder is triggered by the IBM Tivoli Enterprise Console rules, and its logs can be located in $DBDIR/logs/synch/synch_trace.log and synch_msg.log.

Note: On our test Windows IBM Tivoli Enterprise Console server, the logs were created in C:\tmp\itmsynch\logs\synch_tr.

The bin directory in the location $BINDIR/TME/TEC/OM_TEC on UNIX and Windows IBM Tivoli Enterprise Console servers contains some scripts, as shown in Figure 10-22.



```
bash$ pwd
C:/Tivoli/bin/w32-ix86/TME/TEC/OM_TEC/bin
bash$ ls -l
total 80
-rw-rw-rw-   1 0       0         10609 Aug 07 05:58 addrbfile.pl
-rw-rw-rw-   1 0       0          1764 May 13 20:22 itmtec-remove.sh
-rwxrwxrwx   1 0       0           620 Aug 07 05:58 query_state.cmd
-rw-rw-rw-   1 0       0         18243 Aug 07 05:58 sitconfig.sh
-rw-rw-rw-   1 0       0         10452 Aug 07 05:58 sitconfsvruser.sh
-rw-rw-rw-   1 0       0         18784 Aug 07 05:58 situpdate.sh
-rwxrwxrwx   1 0       0           618 Aug 07 05:58 startSUF.cmd
-rwxrwxrwx   1 0       0           667 Aug 07 05:58 stop.cmd
-rwxrwxrwx   1 0       0           619 Aug 07 05:58 test.cmd
-rw-rw-rw-   1 0       0          8595 Aug 07 05:58 upg_sentry_baroc.pl
-rw-rw-rw-   1 0       0          2730 Aug 07 05:58 wcrtitmtask.sh
-rwxrwxrwx   1 0       0          2245 Aug 07 05:58 wrules_check.cmd
bash$ _
```

Figure 10-22 Contents of $BINDIR/TME/TEC/OM_TEC/bin

The script sitconfig.sh is used to configure the following parameters for the event synchronization process:

- The name of the configuration file, which defaults to situpdate.conf.
- The directory location of event cache files.
- The polling interval for updates, which defaults to 3 seconds.
- The number of bytes used to compute the CRC value.
- The buffer flash rate to control the rate at which events will be sent to the TEMS.

Example 10-8 shows the use of this script.

Example 10-8 Usage syntax of sitconfig.sh

```
sitconfig add {[fileName=<conffile>] [fileSize=<size> fileNumber=<num> fi
leLocation=<path> pollingInterval=<sec> crcBytecount=<count> cmsSoapUrl=<url>
bufferFlushRate=<rate> logLevel=<level>]}

sitconfig update{[fileName=<conffile>] [fileSize=<size> fileNumber=<num> fi
leLocation=<path> pollingInterval=<sec> crcBytecount=<count> cmsSoapUrl=<url>
bufferFlushRate=<rate> logLevel=<level>]}
```

```
Where:
       -usage will print its usage statement
       add will create the configuration file.
       update will update the specified configuration file.
       fileName is the name of the configuration file. Default is
situpdate.conf

       fileSize is the maximum size, in bytes, of any single cache file.
Default is 50000.
       fileNumber is the maximum number of cache files that can exist. Default
is 10.
       fileLocation is the directory in which the cache files reside.
       pollingInterval is the number of seconds that the long running program
will sleep when there are no new situation updates to process. Default is 3.
       crcBytecount is the number of bytes that the long running process will
use when it saves the location of the last event it processed.
       cmsSoapUrl is the URL of the Tivoli Enterprise Monitoring Server SOAP
server. Default is cms/soap.
       bufferFlushRate is the rate at which SOAP requests will be sent to the
Tivoli Enterprise Monitoring Server from TEC via Web Services.
       logLevel is the level of debug detailed being provided by the
long-runnin g program.
           The following values are supported :-
              low       Only exceptional debug data will be reported (Default
value)
              med       Intermediate debug data will be reported
              verbose   Detailed debug data will be reported
```

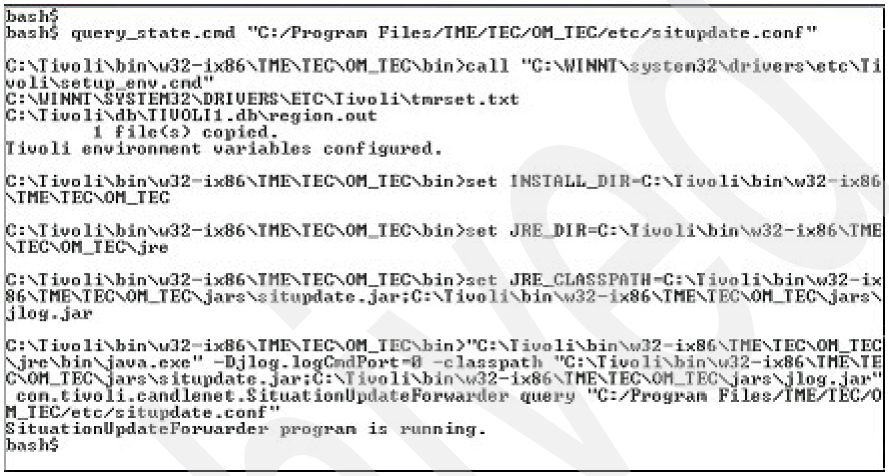
The script sitconfsvuser.sh is used to configure the user ID and password that are used for interaction with the Web services. Example 10-9 shows the use of this script.

Example 10-9 Usage syntax of sitconvsvruser.sh

```
sitconfsvruser {add | update} serverid=<svrid> userid=<userid> password=<
password>
sitconfsvruser delete serverid=<svrid>

sitconfsvruser -usage

Where:
       -usage will print its usage statement
       add will create an entry for the server in the configuration file.
       update will update the server's userid or passwordin the configuration
file.
       delete will delete the server's entry from the specified configuration
file.
```

```
      serverid is the name or IP address of the Tivoli Enterprise Monitoring
Server or hub.
      userid is the name to be used to access the server.
      password is the password to be used to access the server.
```

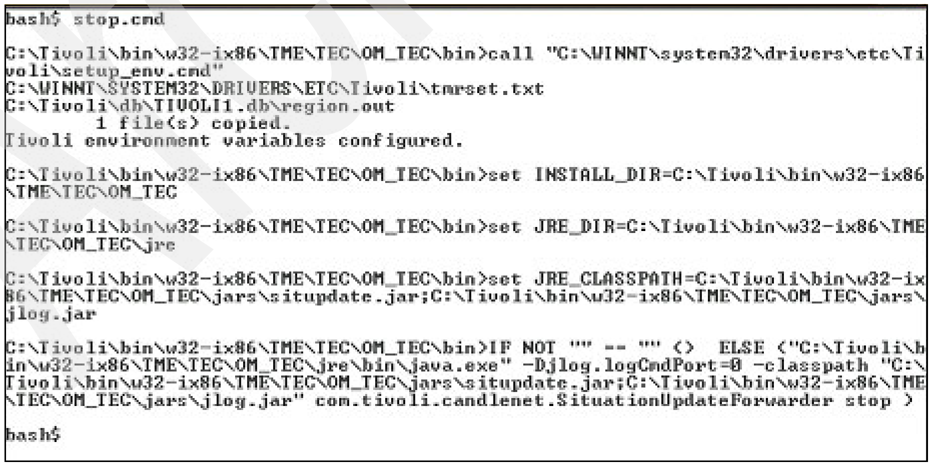
The script **`query_state.cmd`** can be used to query the state of the Situation Update Forwarder. Figure 10-23 shows the use of this script.



```
bash$
bash$ query_state.cmd "C:/Program Files/TME/TEC/OM_TEC/etc/situpdate.conf"

C:\Tivoli\bin\w32-ix86\TME\TEC\OM_TEC\bin>call "C:\WINNT\system32\drivers\etc\Ti
voli\setup_env.cmd"
C:\WINNT\SYSTEM32\DRIVERS\ETC\Tivoli\tmrset.txt
C:\Tivoli\db\TIVOLI1.db\region.out
        1 file(s) copied.
Tivoli environment variables configured.

C:\Tivoli\bin\w32-ix86\TME\TEC\OM_TEC\bin>set INSTALL_DIR=C:\Tivoli\bin\w32-ix86
\TME\TEC\OM_TEC

C:\Tivoli\bin\w32-ix86\TME\TEC\OM_TEC\bin>set JRE_DIR=C:\Tivoli\bin\w32-ix86\TME
\TEC\OM_TEC\jre

C:\Tivoli\bin\w32-ix86\TME\TEC\OM_TEC\bin>set JRE_CLASSPATH=C:\Tivoli\bin\w32-ix
86\TME\TEC\OM_TEC\jars\situpdate.jar;C:\Tivoli\bin\w32-ix86\TME\TEC\OM_TEC\jars\
jlog.jar

C:\Tivoli\bin\w32-ix86\TME\TEC\OM_TEC\bin>"C:\Tivoli\bin\w32-ix86\TME\TEC\OM_TEC
\jre\bin\java.exe" -Djlog.logCmdPort=0 -classpath "C:\Tivoli\bin\w32-ix86\TME\TE
C\OM_TEC\jars\situpdate.jar;C:\Tivoli\bin\w32-ix86\TME\TEC\OM_TEC\jars\jlog.jar"
 com.tivoli.candlenet.SituationUpdateForwarder query "C:/Program Files/TME/TEC/O
M_TEC/etc/situpdate.conf"
SituationUpdateForwarder program is running.
bash$
```

Figure 10-23 Running the query_state.cmd command

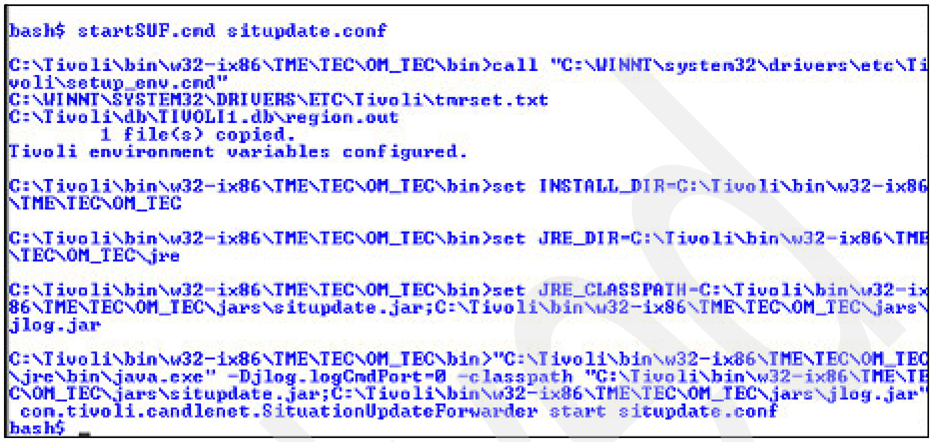
To stop the Situation Update Forwarder, use **`stop.cmd`** (on Windows) or **`stop.sh`** (on UNIX), as shown in Figure 10-24.



```
bash$ stop.cmd

C:\Tivoli\bin\w32-ix86\TME\TEC\OM_TEC\bin>call "C:\WINNT\system32\drivers\etc\Ti
voli\setup_env.cmd"
C:\WINNT\SYSTEM32\DRIVERS\ETC\Tivoli\tmrset.txt
C:\Tivoli\db\TIVOLI1.db\region.out
        1 file(s) copied.
Tivoli environment variables configured.

C:\Tivoli\bin\w32-ix86\TME\TEC\OM_TEC\bin>set INSTALL_DIR=C:\Tivoli\bin\w32-ix86
\TME\TEC\OM_TEC

C:\Tivoli\bin\w32-ix86\TME\TEC\OM_TEC\bin>set JRE_DIR=C:\Tivoli\bin\w32-ix86\TME
\TEC\OM_TEC\jre

C:\Tivoli\bin\w32-ix86\TME\TEC\OM_TEC\bin>set JRE_CLASSPATH=C:\Tivoli\bin\w32-ix
86\TME\TEC\OM_TEC\jars\situpdate.jar;C:\Tivoli\bin\w32-ix86\TME\TEC\OM_TEC\jars\
jlog.jar

C:\Tivoli\bin\w32-ix86\TME\TEC\OM_TEC\bin>IF NOT "" == "" ()  ELSE ("C:\Tivoli\b
in\w32-ix86\TME\TEC\OM_TEC\jre\bin\java.exe" -Djlog.logCmdPort=0 -classpath "C:\
Tivoli\bin\w32-ix86\TME\TEC\OM_TEC\jars\situpdate.jar;C:\Tivoli\bin\w32-ix86\TME
\TEC\OM_TEC\jars\jlog.jar" com.tivoli.candlenet.SituationUpdateForwarder stop )

bash$
```

Figure 10-24 Stopping the Situation Update Forwarder using stop.cmd

Stop the Situation Update Forwarder before you restart the IBM Tivoli Enterprise Console event server. For this, use the script **startSUF.cmd** to start the Situation Update Forwarder, as shown in Figure 10-25.



```
bash$ startSUF.cmd situpdate.conf

C:\Tivoli\bin\w32-ix86\TME\TEC\OM_TEC\bin>call "C:\WINNT\system32\drivers\etc\Ti
voli\setup_env.cmd"
C:\WINNT\SYSTEM32\DRIVERS\ETC\Tivoli\tmrset.txt
C:\Tivoli\db\TIVOLI1.db\region.out
        1 file(s) copied.
Tivoli environment variables configured.

C:\Tivoli\bin\w32-ix86\TME\TEC\OM_TEC\bin>set INSTALL_DIR=C:\Tivoli\bin\w32-ix86
\TME\TEC\OM_TEC

C:\Tivoli\bin\w32-ix86\TME\TEC\OM_TEC\bin>set JRE_DIR=C:\Tivoli\bin\w32-ix86\TME
\TEC\OM_TEC\jre

C:\Tivoli\bin\w32-ix86\TME\TEC\OM_TEC\bin>set JRE_CLASSPATH=C:\Tivoli\bin\w32-ix
86\TME\TEC\OM_TEC\jars\situpdate.jar;C:\Tivoli\bin\w32-ix86\TME\TEC\OM_TEC\jars\
jlog.jar

C:\Tivoli\bin\w32-ix86\TME\TEC\OM_TEC\bin>"C:\Tivoli\bin\w32-ix86\TME\TEC\OM_TEC
\jre\bin\java.exe" -Djlog.logCmdPort=0 -classpath "C:\Tivoli\bin\w32-ix86\TME\TE
C\OM_TEC\jars\situpdate.jar;C:\Tivoli\bin\w32-ix86\TME\TEC\OM_TEC\jars\jlog.jar"
 com.tivoli.candlenet.SituationUpdateForwarder start situpdate.conf
bash$ _
```

Figure 10-25 Starting the Situation Update Forwarder using startSUF.cmd

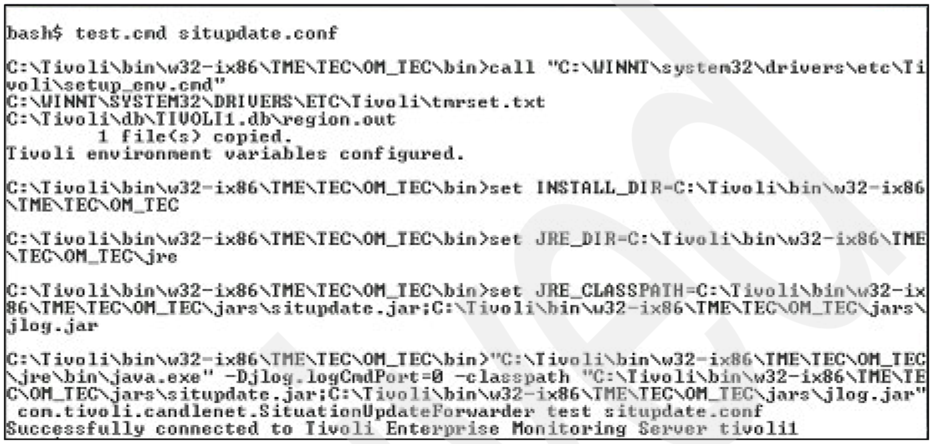
If you specify an incorrect configuration file name, a message will be written in the synch_msg.log file, in a form similar to that in Example 10-10.

Example 10-10 Error message in starting the SUF, shown in synch_msg.log

```
<Message Id="KFAIT0003E" Severity="ERROR">
 <Time Millis="1126522379047"> 2005.09.12 11:52:59.047+01:00</Time>
 <Server Format="IP">tivoli1</Server>
 <ProductId>Monitoring</ProductId>
 <Component>TEC Synchronization</Component>
 <ProductInstance></ProductInstance>
 <LogText><![CDATA[KFAIT0003E The C:\Program
Files\TME\TEC\OM_TEC\etc\C:\Program Files\TME\TEC\OM_TEC\etc\situpdate.conf
(The filename, directory name, or volume label syntax is incorrect)error
message was received while the software attempted to read the configuration
data from C:\Program Files\TME\TEC\OM_TEC\etc\C:/Program
Files/TME/TEC/OM_TEC/etc/situpdate.conf.]]></LogText>
 <Source FileName="com.tivoli.candlenet.SituationUpdateForwarder"
Method="readConfigData"/>
 <TranslationInfo Type="JAVA"
Catalog="com.tivoli.candlenet.messages.TecSynchronizationMessages"
MsgKey="KFAIT0003"><Param><![CDATA[C:\Program
Files\TME\TEC\OM_TEC\etc\C:\Program Files\TME\TEC\OM_TEC\etc\situpdate.conf
(The filename, directory name, or volume label syntax is
incorrect)]]></Param><Param><![CDATA[C:\Program
Files\TME\TEC\OM_TEC\etc\C:/Program
Files/TME/TEC/OM_TEC/etc/situpdate.conf]]></Param></TranslationInfo>
```

```
 <Principal></Principal>
</Message>
```

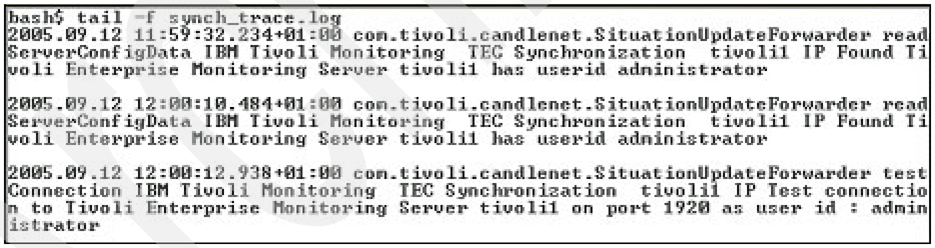
The script **`test.cmd`** can be used to test connectivity between the Situation Update Forwarder and the TEMS, as shown in Figure 10-26.



```
bash$ test.cmd situpdate.conf

C:\Tivoli\bin\w32-ix86\TME\TEC\OM_TEC\bin>call "C:\WINNT\system32\drivers\etc\Ti
voli\setup_env.cmd"
C:\WINNT\SYSTEM32\DRIVERS\ETC\Tivoli\tmrset.txt
C:\Tivoli\db\TIVOLI1.db\region.out
        1 file(s) copied.
Tivoli environment variables configured.

C:\Tivoli\bin\w32-ix86\TME\TEC\OM_TEC\bin>set INSTALL_DIR=C:\Tivoli\bin\w32-ix86
\TME\TEC\OM_TEC

C:\Tivoli\bin\w32-ix86\TME\TEC\OM_TEC\bin>set JRE_DIR=C:\Tivoli\bin\w32-ix86\TME
\TEC\OM_TEC\jre

C:\Tivoli\bin\w32-ix86\TME\TEC\OM_TEC\bin>set JRE_CLASSPATH=C:\Tivoli\bin\w32-ix
86\TME\TEC\OM_TEC\jars\situpdate.jar;C:\Tivoli\bin\w32-ix86\TME\TEC\OM_TEC\jars\
jlog.jar

C:\Tivoli\bin\w32-ix86\TME\TEC\OM_TEC\bin>"C:\Tivoli\bin\w32-ix86\TME\TEC\OM_TEC
\jre\bin\java.exe" -Djlog.logCmdPort=0 -classpath "C:\Tivoli\bin\w32-ix86\TME\TE
C\OM_TEC\jars\situpdate.jar;C:\Tivoli\bin\w32-ix86\TME\TEC\OM_TEC\jars\jlog.jar"
 com.tivoli.candlenet.SituationUpdateForwarder test situpdate.conf
Successfully connected to Tivoli Enterprise Monitoring Server tivoli1
```

Figure 10-26 Testing the connectivity between the SUF and TEMS using test.cmd

The test.cmd script also appends some lines in the synch_trace.log, as shown in Figure 10-27.



```
bash$ tail -f synch_trace.log
2005.09.12 11:59:32.234+01:00 com.tivoli.candlenet.SituationUpdateForwarder read
ServerConfigData IBM Tivoli Monitoring  TEC Synchronization  tivoli1 IP Found Ti
voli Enterprise Monitoring Server tivoli1 has userid administrator

2005.09.12 12:00:10.484+01:00 com.tivoli.candlenet.SituationUpdateForwarder read
ServerConfigData IBM Tivoli Monitoring  TEC Synchronization  tivoli1 IP Found Ti
voli Enterprise Monitoring Server tivoli1 has userid administrator

2005.09.12 12:00:12.938+01:00 com.tivoli.candlenet.SituationUpdateForwarder test
Connection IBM Tivoli Monitoring  TEC Synchronization  tivoli1 IP Test connectio
n to Tivoli Enterprise Monitoring Server tivoli1 on port 1920 as user id : admin
istrator
```

Figure 10-27 Contents of synch_trace.log appended to by test.cmd

The script addrbfile.pl can be used to add the omegemon event definitions and rules to a new or existing rulebase. Example 10-11 shows its usage.

Example 10-11 Usage of addrbfile.pl

```
bash$ ./addrbfile.pl
Usage:
    addrbfile.pl -new rbNname {-r|-b} filePath [-copy fromRb]

    addrbfile.pl -addto rbNname {-r|-b} filePath
```

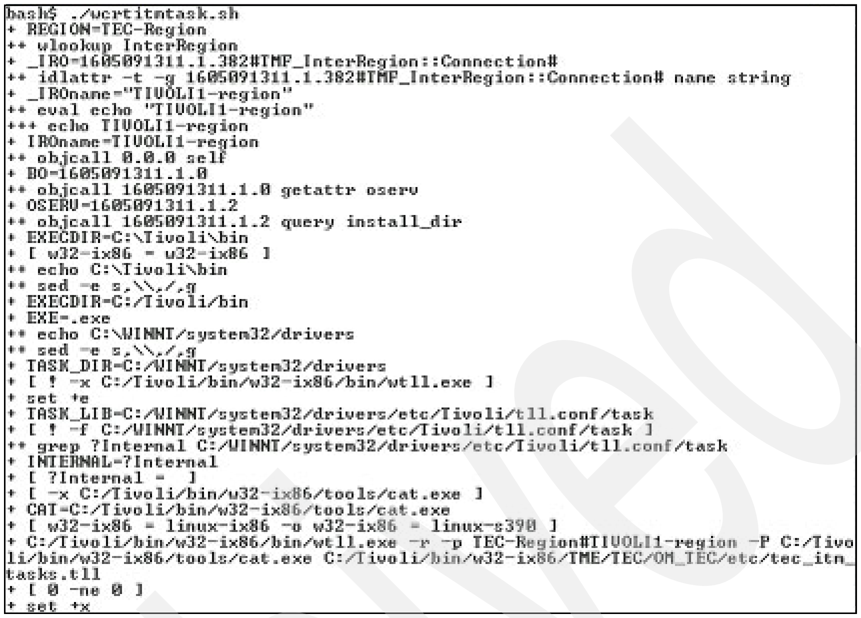
```
Where:
        -new     will be used to create a new rulebase.
        -addto   will be used to add the files to an existing rulebase.
        rbName   is the full path of the rulebase to be created or updated.
        -r       will be used to specify the ruleset file to be imported.
        -b       will be used to specify the baroc file to be imported.
        filePath is the full path and name of the file to be copied into the
rulebase.
        -copy    will be used to copy an existing rulebase into the new
rulebase.
        fromRb   is the name of the rulebase that will be copied into the new
rulebase.
```

The **wrules_check** command provides you with the ability to assess the impact on an existing set of rules whenever an event class design must be changed. Use this command to verify which rules may have been affected by these event class definition changes. This command is packaged with IBM Tivoli Monitoring 6.1 and will be provided with the next release of IBM Tivoli Enterprise Console, as its function is not specific to IBM Tivoli Monitoring. Example 10-12 shows the usage syntax of this command.

Example 10-12 Usage syntax of wrules_check.cmd

```
Usage: wrules_check {[-h] | [-v] | [-rd rules_directory] [-cd
baroc_classes_directory]
{class[,attribute,attribute...][:classN,attributeN,attributeN] | [-f clas
s_file]} [-of output_file]}
Where:
               -h  display this help and exit
               -v  display the rule check utility version and exit
               -f  file with classes and attributes to be checked
               -rd rule base directory. current rule base directory if not
provided
               -cd BAROC class directory. current BAROC classes directory if
not provided
               -of rule check output file
```

Figure 10-28 shows example output of running **`wrules_check.cmd`**.



```
bash$ ./wcrtitmtask.sh
+ REGION=TEC-Region
++ wlookup InterRegion
+ _IRO=1605091311.1.382#TMF_InterRegion::Connection#
++ idlattr -t -g 1605091311.1.382#TMF_InterRegion::Connection# name string
+ _IROname="TIVOLI1-region"
++ eval echo "TIVOLI1-region"
+++ echo TIVOLI1-region
+ IROname=TIVOLI1-region
++ objcall 0.0.0 self
+ BO=1605091311.1.0
++ objcall 1605091311.1.0 getattr oserv
+ OSERV=1605091311.1.2
++ objcall 1605091311.1.2 query install_dir
+ EXECDIR=C:\Tivoli\bin
+ [ w32-ix86 = w32-ix86 ]
++ echo C:\Tivoli\bin
++ sed -e s,\\,/,g
+ EXECDIR=C:/Tivoli/bin
+ EXE=.exe
++ echo C:\WINNT/system32/drivers
++ sed -e s,\\,/,g
+ TASK_DIR=C:/WINNT/system32/drivers
+ [ ! -x C:/Tivoli/bin/w32-ix86/bin/wtll.exe ]
+ set +e
+ TASK_LIB=C:/WINNT/system32/drivers/etc/Tivoli/tll.conf/task
+ [ ! -f C:/WINNT/system32/drivers/etc/Tivoli/tll.conf/task ]
++ grep ?Internal C:/WINNT/system32/drivers/etc/Tivoli/tll.conf/task
+ INTERNAL=?Internal
+ [ ?Internal =  ]
+ [ -x C:/Tivoli/bin/w32-ix86/tools/cat.exe ]
+ CAT=C:/Tivoli/bin/w32-ix86/tools/cat.exe
+ [ w32-ix86 = linux-ix86 -o w32-ix86 = linux-s390 ]
+ C:/Tivoli/bin/w32-ix86/bin/wtll.exe -r -p TEC-Region#TIVOLI1-region -P C:/Tivo
li/bin/w32-ix86/tools/cat.exe C:/Tivoli/bin/w32-ix86/TME/TEC/OM_TEC/etc/tec_itm_
tasks.tll
+ [ 0 -ne 0 ]
+ set +x
```

Figure 10-28 Running wcrtitmtask.sh

The script wcrtitmtask.sh creates the TEC ITM Synchronization Tasks as shown in Figure 10-28. These tasks are described in 10.7, "Synchronizing IBM Tivoli Monitoring situations and Tivoli Enterprise Console events" on page 627.

Example 10-13 Sample output when running wrules_check.cmd

```
wrules_check.cmd Omegamon_Base

Loading Classes
Parsing C:\rb\itm61\itm61\TEC_CLASSES\root.baroc
Parsing C:\rb\itm61\itm61\TEC_CLASSES\tec.baroc
...
...
Parsing C:\rb\itm61\itm61\TEC_CLASSES\Sentry.baroc
Parsing C:\rb\itm61\itm61\TEC_CLASSES\omegamon.baroc

Loading Rule Sets

Parsing C:/rb/itm61/itm61/TEC_RULES\maintenance_mode.rls

...

************************************************************
```

```
Parsing C:/rb/itm61/itm61/TEC_RULES\maintenance_mode.rls

**********************************************************
...
Parsing C:/rb/itm61/itm61/TEC_RULES\omegamon.rls

rules impacted by class:
Omegamon_Base:candle_synchronization,flush_om_updates,master_reset,resync_reset
_
events,update,update_Y_status
TEC_ITM_OM_Situation_Sync_Error:process_sit_sync_error

rules impacted by outside operator:
Omegamon_Base:process_sit_events_only
```

10.6 Integrating the Tivoli Enterprise Console in the TEP workspace

The Tivoli Enterprise Portal workspaces can be configured to provide various views into a monitored environment. This consolidated monitoring portal can provide a view into the Event Management data through the use of the *TEC View*.

The TEC View allows for access to the Java Tivoli Enterprise Console within a workspace. This view has the same look and feel as the Tivoli Enterprise Console Java Console and has a dynamic query filter based on IP address, host names, and managed system names. When the user first adds the TEC View to a workspace, the filters are determined.

As well as launching Tivoli Framework Tasks, events can be viewed, acknowledged, and closed within this TEC View.

There is also built-in integration with the NetView Console, which can be launched from within the TEC View.

Important: Although the event processing capabilities of IBM Tivoli Monitoring 6.1 are very powerful for a monitoring product, it was not designed to be an event consolidation Manager of Managers, such as IBM Tivoli Enterprise Console. Tivoli Enterprise Console is where IBM will continue to invest in a very high-volume, heterogeneous, event integration, and persistence technology. The IBM direction is to make the basic event processing capabilities of IBM Tivoli Monitoring 6.1 fit well and naturally into those of the Tivoli Enterprise Console and to share technology between the two products to make this integration helpful and natural to the client's usage scenarios.

10.6.1 Adding TEC Views to your workspace

The toolbar in the Tivoli Enterprise Portal has a Tivoli Enterprise Console icon, as shown in Figure 10-29.

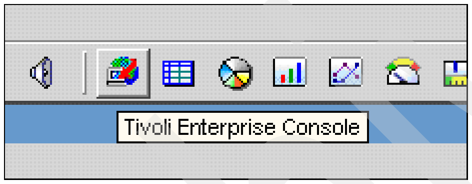
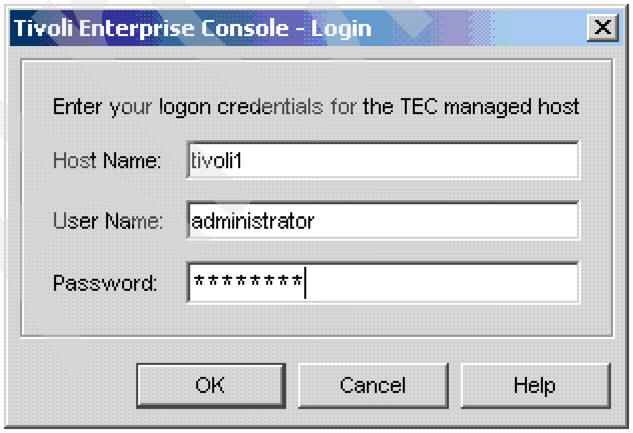


Figure 10-29 TEC View icon in the TEP toolbar

When this icon is selected and added to your workspace, you will be prompted to enter the Tivoli Enterprise Console server credentials (Figure 10-30).



Tivoli Enterprise Console - Login

Enter your logon credentials for the TEC managed host

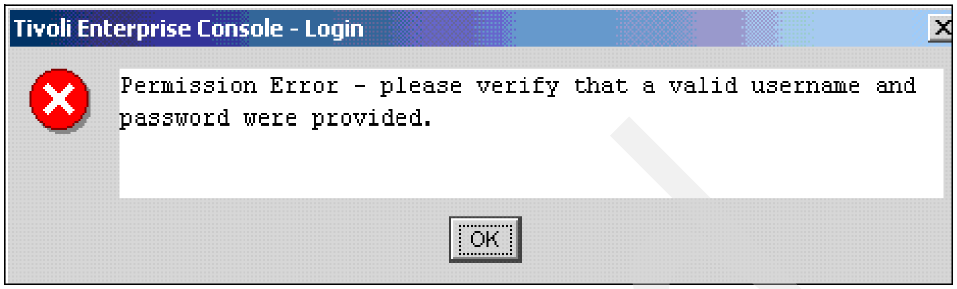
Host Name: tivoli1

User Name: administrator

Password: ********

Figure 10-30 Enter Tivoli Enterprise Console server credentials

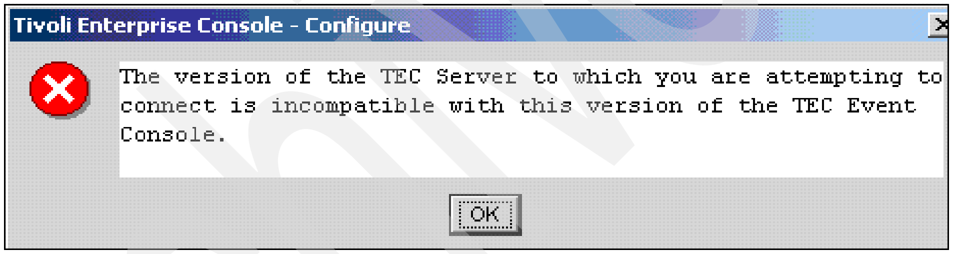
If you entered incorrect details, the error shown in Figure 10-31 will be displayed.



Permission Error - please verify that a valid username and password were provided.

Figure 10-31 Incorrect IBM Tivoli Enterprise Console server details entered

The TEC View requires at least IBM Tivoli Enterprise Console 3.9 with Fix Pack 3 installed. If the incorrect version of the IBM Tivoli Enterprise Console server is discovered by the Tivoli Enterprise Portal, then the error shown in Figure 10-32 will be displayed.



The version of the TEC Server to which you are attempting to connect is incompatible with this version of the TEC Event Console.

Figure 10-32 Incorrect IBM Tivoli Enterprise Console server version discovered

Otherwise you are prompted to provide filtering details as shown in Figure 10-33.

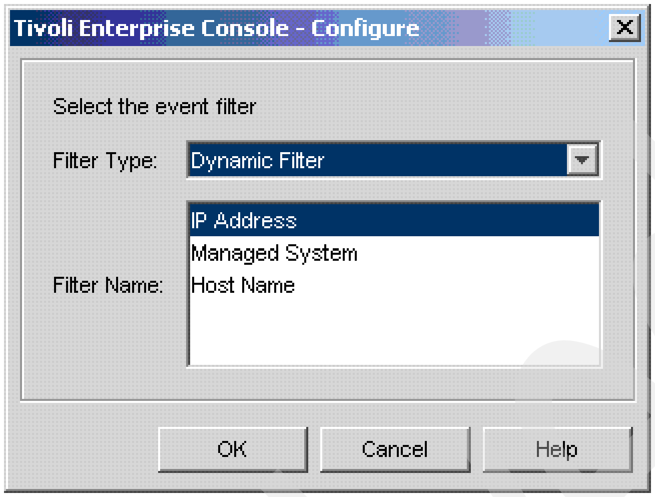


Figure 10-33 Tivoli Enterprise Console event filtering options

A single workspace can have multiple TEC Event Viewer windows set up with different event filters. The dynamic filter is context-sensitive based on the navigation tree. The highlighted entry in the navigation view determines which events will be displayed; for example:

- To view all events, create the event view on the enterprise level.
- To view all UNIX events, create the event view on the agent type level UNIX.
- To view only events from a single system, create the event view on the agent level.

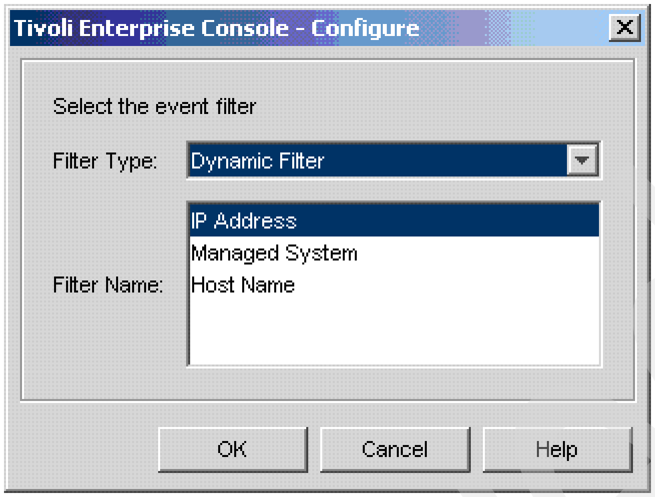


Figure 10-34 Tivoli Enterprise Console event filtering options

As shown in Figure 10-35, the Filter Type pull-down list shows all event groups that have been defined in your IBM Tivoli Enterprise Console server configuration. If you require new groups, you must define them using the Java Tivoli Enterprise Console.

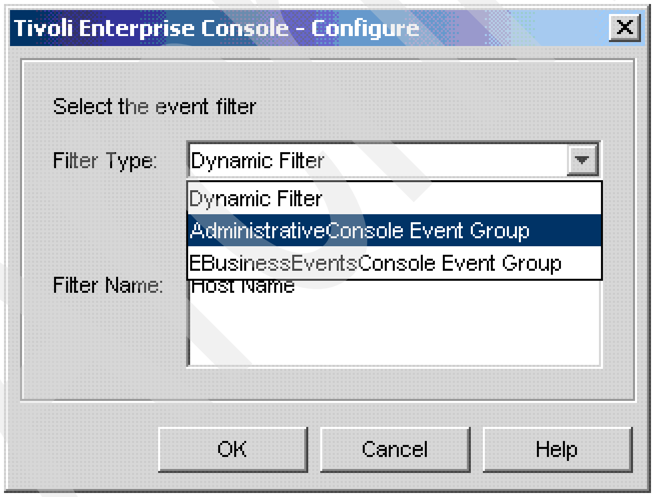
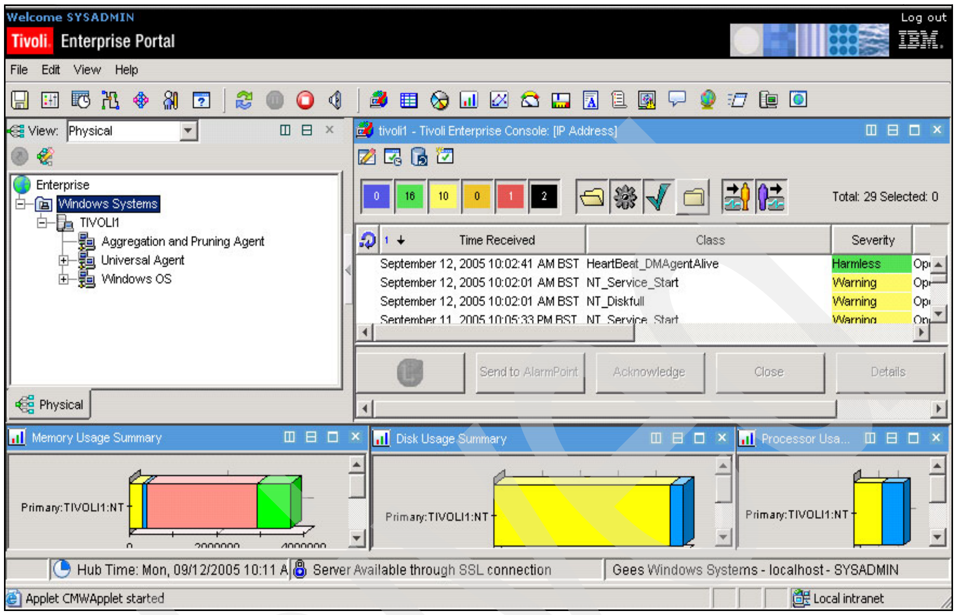

Tivoli Enterprise Console - Configure
Select the event filter
Filter Type:
Dynamic Filter
Dynamic Filter
AdministrativeConsole Event Group
EBusinessEventsConsole Event Group
Filter Name:
Host Name
OK
Cancel
Help

Figure 10-35 Tivoli Enterprise Console event filtering options

Click **OK** to insert the TEC view into your workspace (Figure 10-36).



Total: 29 Selected: 0

Time Received	Class	Severity	
September 12, 2005 10:02:41 AM BST	HeartBeat_DMAgentAlive	Harmless	Op
September 12, 2005 10:02:01 AM BST	NT_Service_Start	Warning	Op
September 12, 2005 10:02:01 AM BST	NT_Diskfull	Warning	Op
September 11, 2005 10:05:33 PM BST	NT_Service_Start	Warning	Op

Hub Time: Mon, 09/12/2005 10:11 A Server Available through SSL connection Gees Windows Systems - localhost - SYSADMIN

Applet CMWApplet started Local intranet

Figure 10-36 TEP workspace with TEC View, showing TEC events

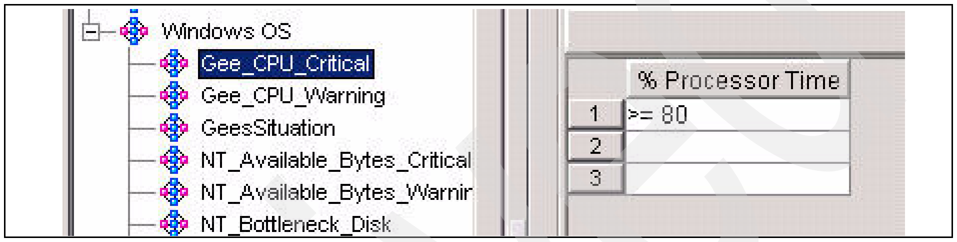
Events can be viewed, acknowledged, and closed in the same way as if you were using the Java Tivoli Enterprise Console.

Note: No event console configuration can be done using the TEP.

10.6.2 Receiving IBM Tivoli Enterprise Console events

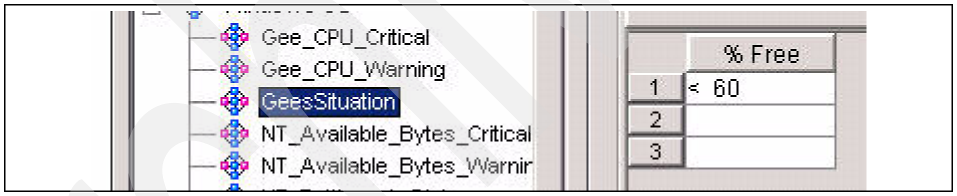
After you have configured the OTEA as described in 10.2, “OMEGAMON IBM Tivoli Enterprise Console Event Adapter (OTEA)” on page 585, you should start to receive IBM Tivoli Enterprise Console events triggered by your situations. We now offer a simple scenario. For more information about how to create situations and best practices, refer to Chapter 8, “Real-life scenarios” on page 469.

1. Figure 10-37 and Figure 10-38 show some simple situations. First we define a situation that triggers when the % Processor time is >= 80% and another situation that triggers when the % Disk Space free is < 60 %.



	% Processor Time
1	>= 80
2	
3	

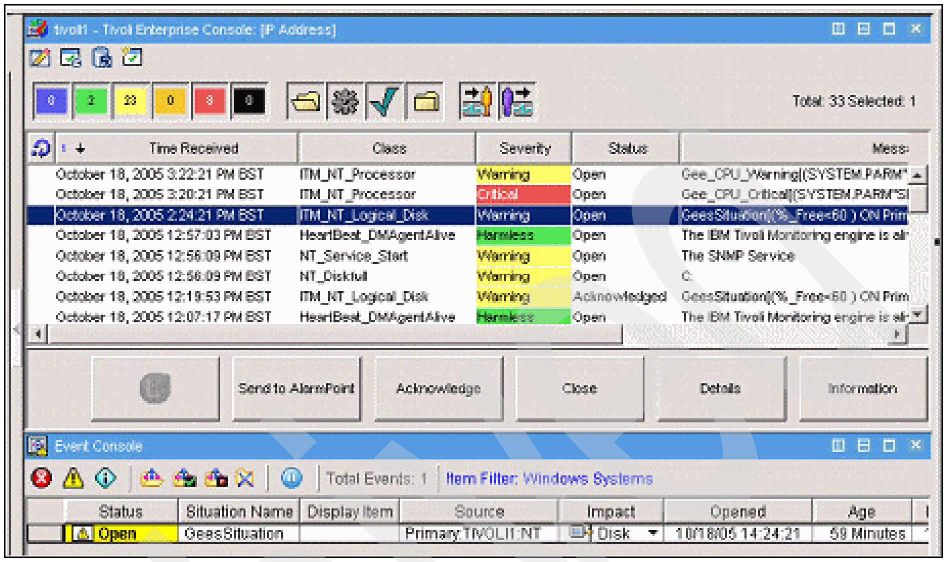
Figure 10-37 Situation that triggers when % Processor time >= 80%



	% Free
1	< 60
2	
3	

Figure 10-38 Situation that triggers when % Disk Space free < 60%

2. These situations are applied to the agent, nodes that are running the Windows OS Agent. When the CPU load is increased, we start to receive events, as shown in Figure 10-39.



tivoli1 - Tivoli Enterprise Console: [IP Address]

Total: 33 Selected: 1

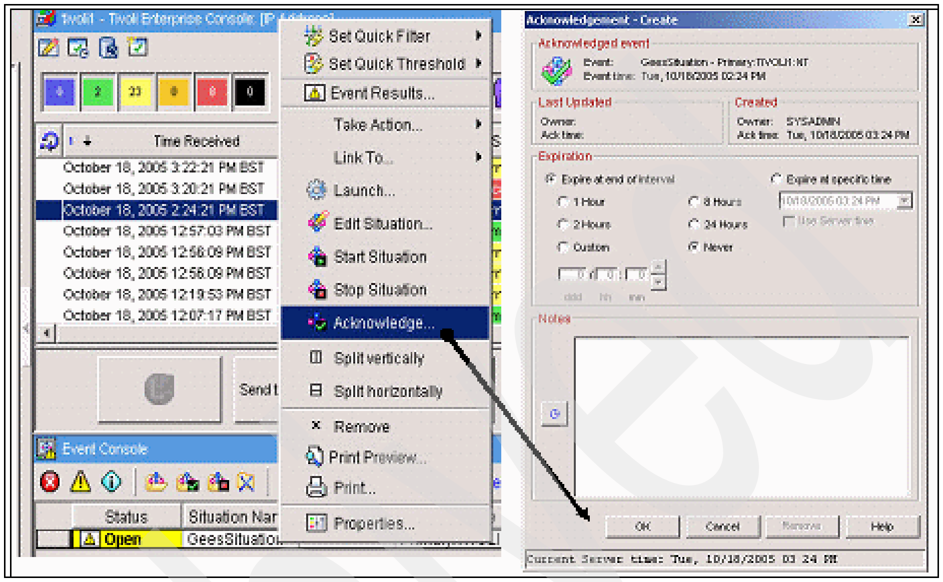
Time Received	Class	Severity	Status	Mess
October 18, 2005 3:22:21 PM BST	ITM_NT_Processor	Warning	Open	Gee_CPU_Warning[(SYSTEM.PARM"
October 18, 2005 3:20:21 PM BST	ITM_NT_Processor	Critical	Open	Gee_CPU_Critical[(SYSTEM.PARM"SI
October 18, 2005 2:24:21 PM BST	ITM_NT_Logical_Disk	Warning	Open	GeesSituation[(%_Free<60) ON Prim
October 18, 2005 12:57:03 PM BST	HeartBeat_DMAgentAlive	Harmless	Open	The IBM Tivoli Monitoring engine is alr
October 18, 2005 12:56:09 PM BST	NT_Service_Start	Warning	Open	The SNMP Service
October 18, 2005 12:56:09 PM BST	NT_Diskfull	Warning	Open	C:
October 18, 2005 12:19:53 PM BST	ITM_NT_Logical_Disk	Warning	Acknowledged	GeesSituation[(%_Free<60) ON Prim
October 18, 2005 12:07:17 PM BST	HeartBeat_DMAgentAlive	Harmless	Open	The IBM Tivoli Monitoring engine is alr

Total Events: 1 Item Filter: Windows Systems

Status	Situation Name	Display Item	Source	Impact	Opened	Age
Open	GeesSituation		Primary:TIVOLI1:NT	Disk	10/18/05 14:24:21	59 Minutes

Figure 10-39 A situation triggering a IBM Tivoli Enterprise Console event

3. In this example, the situation GeesSitutation is open, and causes an ITM_NT_Logical_Disk event to be sent. This situation is acknowledged by right-clicking it and selecting **Acknowledge** as shown in Figure 10-40.



Time Received
October 18, 2005 3:22:21 PM BST
October 18, 2005 3:20:21 PM BST
October 18, 2005 2:24:21 PM BST
October 18, 2005 12:57:03 PM BST
October 18, 2005 12:56:09 PM BST
October 18, 2005 12:56:09 PM BST
October 18, 2005 12:19:53 PM BST
October 18, 2005 12:07:17 PM BST

Status	Situation Nar
Open	GeesSituatio

Acknowledgement - Create

Acknowledged event
Event: GeesSituation - Primary:TIVOLI1:NT
Event time: Tue, 10/18/2005 02:24 PM

Last Updated
Owner:
Ack time:

Created
Owner: SYSADMIN
Ack time: Tue, 10/18/2005 03:24 PM

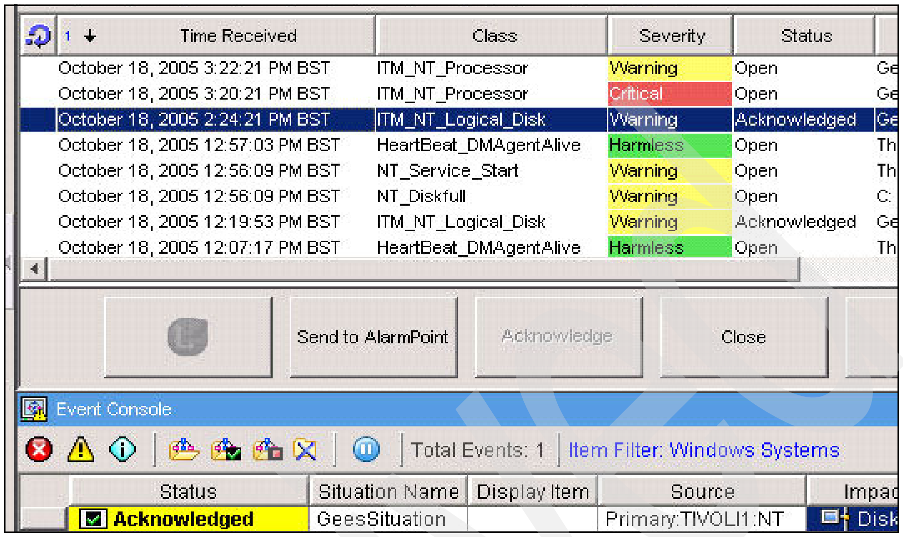
Expiration
[✓ Expire at end of interval] Expire at specific time
1 Hour 8 Hours
2 Hours 24 Hours
Custom [✓ Never]
10/18/2005 03:24 PM
Use Server time
0 d 0 : 0
ddd hh mm

Notes

Current Server time: Tue, 10/18/2005 03 24 PM

Figure 10-40 Acknowledging a situation

4. The corresponding IBM Tivoli Enterprise Console event is then acknowledged automatically, as shown in Figure 10-41.



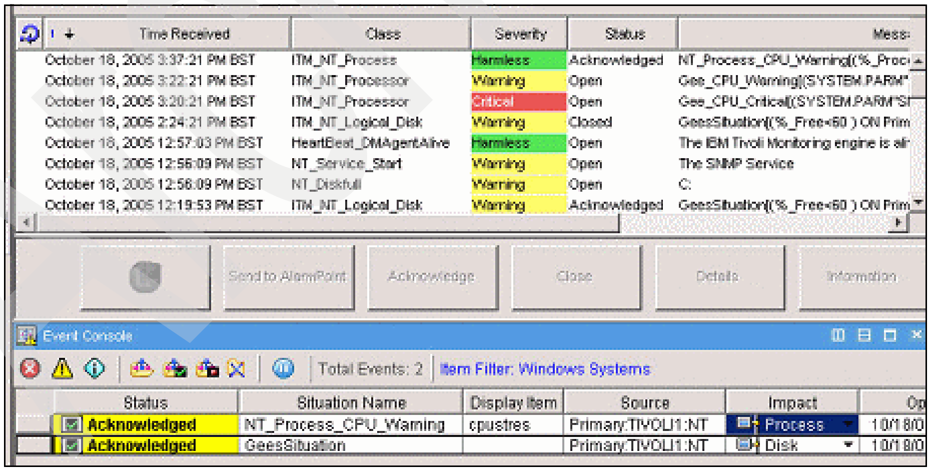
Time Received	Class	Severity	Status	
October 18, 2005 3:22:21 PM BST	ITM_NT_Processor	Warning	Open	Ge
October 18, 2005 3:20:21 PM BST	ITM_NT_Processor	Critical	Open	Ge
October 18, 2005 2:24:21 PM BST	ITM_NT_Logical_Disk	Warning	Acknowledged	Ge
October 18, 2005 12:57:03 PM BST	HeartBeat_DMAgentAlive	Harmless	Open	Th
October 18, 2005 12:56:09 PM BST	NT_Service_Start	Warning	Open	Th
October 18, 2005 12:56:09 PM BST	NT_Diskfull	Warning	Open	C:
October 18, 2005 12:19:53 PM BST	ITM_NT_Logical_Disk	Warning	Acknowledged	Ge
October 18, 2005 12:07:17 PM BST	HeartBeat_DMAgentAlive	Harmless	Open	Th

Total Events: 1 Item Filter: Windows Systems

Status	Situation Name	Display Item	Source	Impac
Acknowledged	GeesSituation		Primary:TIVOLI1:NT	Disk

Figure 10-41 Situation status change synchronizes the TEC event

5. As the CPU load on the agent is increased, a situation is triggered and an event of type ITM_NT_Process is received. When this event is acknowledged, the corresponding situation is also acknowledged as shown in Figure 10-42.



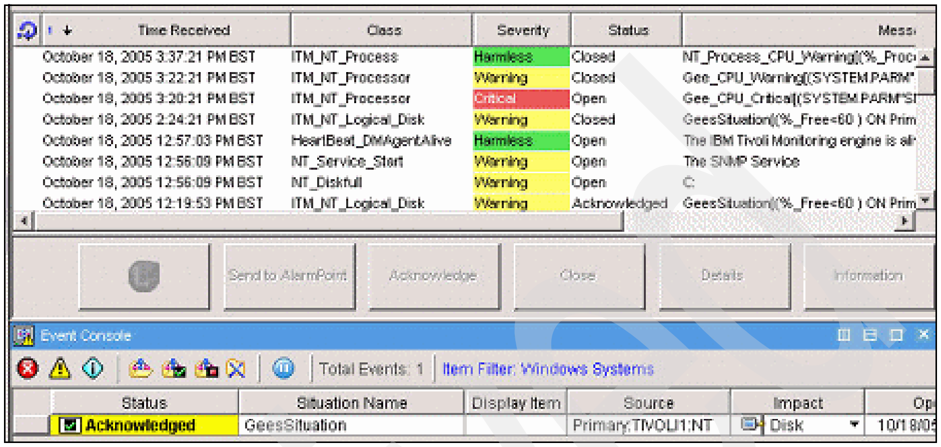
Time Received	Class	Severity	Status	Mess:
October 18, 2005 3:37:21 PM BST	ITM_NT_Process	Harmless	Acknowledged	NT_Process_CPU_Warning[(%_Proc
October 18, 2005 3:22:21 PM BST	ITM_NT_Processor	Warning	Open	Gee_CPU_Warning[(SYSTEM.PARM"
October 18, 2005 3:20:21 PM BST	ITM_NT_Processor	Critical	Open	Gee_CPU_Critical[(SYSTEM.PARM"SI
October 18, 2005 2:24:21 PM BST	ITM_NT_Logical_Disk	Warning	Closed	GeesSituation[(%_Free<60) ON Prim
October 18, 2005 12:57:03 PM BST	HeartBeat_DMAgentAlive	Harmless	Open	The IBM Tivoli Monitoring engine is al
October 18, 2005 12:56:09 PM BST	NT_Service_Start	Warning	Open	The SNMP Service
October 18, 2005 12:56:09 PM BST	NT_Diskfull	Warning	Open	C:
October 18, 2005 12:19:53 PM BST	ITM_NT_Logical_Disk	Warning	Acknowledged	GeesSituation[(%_Free<60) ON Prim

Total Events: 2 Item Filter: Windows Systems

Status	Situation Name	Display Item	Source	Impact	Op
Acknowledged	NT_Process_CPU_Warning	cpustres	Primary:TIVOLI1:NT	Process	10/18/0
Acknowledged	GeesSituation		Primary:TIVOLI1:NT	Disk	10/18/0

Figure 10-42 New TEC event arrives and is acknowledged

6. When the CPU load is reduced, the NT_Process_CPU_Warning situation automatically closes, and this causes the ITM_NT_Process event to close automatically, as shown in Figure 10-43.



Time Received	Class	Severity	Status	Mess
October 18, 2005 3:37:21 PM BST	ITM_NT_Process	Harmless	Closed	NT_Process_CPU_Warning[(%_Proc
October 18, 2005 3:22:21 PM BST	ITM_NT_Processor	Warning	Closed	Gee_CPU_Warning[(SYSTEM.PARM"
October 18, 2005 3:20:21 PM BST	ITM_NT_Processor	Critical	Open	Gee_CPU_Critical[(SYSTEM.PARM"SI
October 18, 2005 2:24:21 PM BST	ITM_NT_Logical_Disk	Warning	Closed	GeesSituation[(%_Free<60) ON Prim
October 18, 2005 12:57:03 PM BST	HeartBeat_DMAgentAlive	Harmless	Open	The IBM Tivoli Monitoring engine is al
October 18, 2005 12:56:09 PM BST	NT_Service_Start	Warning	Open	The SNMP Service
October 18, 2005 12:56:09 PM BST	NT_Diskfull	Warning	Open	C:
October 18, 2005 12:19:53 PM BST	ITM_NT_Logical_Disk	Warning	Acknowledged	GeesSituation[(%_Free<60) ON Prim

Total Events: 1 | Item Filter: Windows Systems

Status	Situation Name	Display Item	Source	Impact	Op
Acknowledged	GeesSituation		Primary:TIVOLI1:NT	Disk	10/18/05

Figure 10-43 Situation closed, which closes IBM Tivoli Enterprise Console event

10.7 Synchronizing IBM Tivoli Monitoring situations and Tivoli Enterprise Console events

Updates to events can be triggered in a number of ways:

- A Tivoli Enterprise Console rule
- Acknowledging or closing an event in the Tivoli Enterprise Console
- An update event send by the Tivoli Enterprise Monitoring Server
- A reset of the TEMS
- When the TEMS goes into Hot Standby

The tables in the next section show various scenarios that can cause event updates.

10.7.1 Event updates initiated by the TEMS

The first column of Table 10-2 on page 628 shows actions that are initiated from the TEMS. The synchronization direction is from the TEMS to the IBM Tivoli Enterprise Console.

Table 10-2 Event Updates: TEMS → IBM Tivoli Enterprise Console

Action	TEMS status	ITM status	Tivoli Enterprise Console status	Comment
situation=true	open situation event	Y	open event	
situation=false	close situation event	N	close event	
situation deleted	close situation event	D	close event	
operator stops situation in TEPS	stopped situation	P Y → P	keep event status	only if event exists
operator acks situation in TEPS	situation acknowledged	A	acknowledge event	
operator resurface situation in TEPS	resurface ack situation event	F A → Y	ack event - → open event open event if status is not ack (synch problem)	This works for sit_resurface_def _action= ACCEPT (default) status change allowed
situation ack expired	resurface ack situation event	E A → Y	ack event - → open event open event if status is not ack (synch problem)	This works for sit_resurface_def _action= ACCEPT sit_resurface_def _action= REJECT (default) status change not allowed

10.7.2 Event updates initiated from the Tivoli Enterprise Console server

The actions shown in the first column are initiated from the Tivoli Enterprise Console server. The synchronization direction is from the Tivoli Enterprise Console to the TEMS.

Table 10-3 Event updates: Tivoli Enterprise Console → TEMS

Action	Tivoli Enterprise Console status	TEMS	ITM status	Comment
operator / TEC rule acknowledges event	acknowledge event	acknowledge event	A	
TEC rule reopens acknowledges event at Tivoli Enterprise Console	reopen event	resurface acknowledge event	Y	This works for sit_resurface_def_action= ACCEPT sit_resurface_def_action= REJECT (default) status change not allowed
operator / TEC rule closes event	close event	close event	N	No reevaluation of the situation

10.7.3 Event updates initiated by a TEMS Recycle Master Event

The actions shown in the first column, are initiated from the TEMS when it has been restarted. The synchronization direction is from the TEMS to the Tivoli Enterprise Console.

Table 10-4 Event updates: TEMS Recycle → Tivoli Enterprise Console

Action	**TEMS status**	**Tivoli Enterprise Console status**	**Comment**
TEMS Stop	all situations disappear		Nothing happens before the TEMS is active again
TEMS active again	master reset event with situation_ name="**" new situation events opened after recycle	master_reset_fla g=”R” in all ITM events open events	All open and acknowledged events of the recycled TEMS are flagged.
Tivoli Enterprise Console task: close selected situation events		close selected events of the recycled TEMS	No other action than close possible.
Tivoli Enterprise Console task: close all situation events		close all events of the recycled TEMS	No other action than close possible.
Tivoli Enterprise Console task: resynch situation events		open events with "R" and open new situation events after recycle	Compares new with old event (situation_name, situation_origin, situation_displayitem). If matches are found, the old event is linked to new one and then closed. No 100% guarantee. Possible to change this rule, for example, taking over trouble ticket id (sample provided as template).

10.7.4 Event updates initiated by a TEMS Hot Standby Master Event

The actions shown in the first column are initiated from the TEMS when it switches to the standby TEMS. The synchronization direction is from the TEMS to the Tivoli Enterprise Console.

If the hub monitoring server fails, hub functions automatically switch to the standby monitoring server. IBM Tivoli Monitoring automatically connects all remote monitoring servers and agents to the standby monitoring server. No automatic switch returns control to the hub monitoring server when it is available. If you want to switch back to the hub monitoring server, you must manually start the hub monitoring server and stop the standby monitoring server.

Table 10-5 Event Updates: TEMS Hot Standby → Tivoli Enterprise Console

Action	**TEMS status**	**Tivoli Enterprise Console status**	**Comment**
TEMS stop	all situations disappear		Nothing happens before the TEMS is active again.
TEMS active again (standby)	master reset event with situation_ name="**" master_reset_ flag = 'S' situation_origi n=<old TEMS name> new situation events opened after recycle	master_reset_flag ="R" in all ITM events open events	All open and acknowledged events of the TEMS that has shutdown are flagged and the cms_hostname is changed to the new primary TEMS.
Tivoli Enterprise Console task: close selected situation events		close events of the recycled TEMS	No action other than close possible.
Tivoli Enterprise Console task: close all situation events		close all events of the recycled TEMS	No action other than close possible.

Action	**TEMS status**	**Tivoli Enterprise Console status**	**Comment**
Tivoli Enterprise Console task: resynch situation events		open events with "R" and open new situation events after recycle	Compares new with old event (situation_name, situation_origin, situation_displayitem). If matches are found, the old event is linked to new one and then closed. No 100% guarantee. Possible to change this rule, for example, taking over trouble ticket ID (sample provided as template).

11

Agent deployment using IBM Tivoli Configuration Manager Software Distribution

As we saw in 2.4.2, “Tivoli Configuration Manager V4.2” on page 54, it is possible to deploy Tivoli Monitoring Agents using Tivoli Configuration Manager. Because many IBM Tivoli Software customers already have an investment in Tivoli Management Framework 4.1.1 and Tivoli Configuration Manager 4.2, it certainly makes sense for them to use Tivoli Configuration Manager for this purpose as well. In this chapter, we talk about how you can use Tivoli Configuration Manager 4.2 capabilities for large-scale deployment of TEMAs.

This chapter includes these topics:

- Deployment of TEMAs using IBM Tivoli Configuration Manager
- Prerequisites
- SPDs developed with this solution
- Configuring IBM Tivoli Monitoring 6.1 Agent SPDs
- Deploying IBM Tivoli Monitoring 6.1 Agents SPDs

11.1 Deployment of TEMAs using IBM Tivoli Configuration Manager

IBM Tivoli Monitoring Agents can be installed through a setup program that can be run in attended or unattended (silent) mode or using the command line. The deployment mechanism makes use of a descriptor (.dsc) file that contains all of the information required to deploy, install, and configure an agent.

You can also use Tivoli Configuration Manager's robust remote deployment capabilities to deploy IBM Tivoli Monitoring Agents. To facilitate this process, we provide software package definition files (SPDs) that you can use for every existing agent. If you are already using Tivoli Configuration Manager for software deployment, it certainly makes sense to use it for the deployment of IBM Tivoli Monitoring Agents and combine this with the rest of your Change Management process. This provides all of the benefits of Tivoli Configuration Manager such as bandwidth control, centralized monitoring of the deployment, ability to set prerequisite packages, integration with Inventory database, ability to perform user-initiated install (pull versus the push method), and so forth.

The TEMA deployment solution using Tivoli Configuration Manager is comprised of one SPD file for each available IBM Tivoli Monitoring 6.1 Agent and a set of files containing silent response files and scripts used by the install and uninstall processes.

Note: The SPD is a text file that describes a software package. It contains a list of actions to be executed on the target and can be updated with a text editor or the Software Package Editor. Files are collected from the source host at distribution time, as opposed to software package blocks (SPB), format, where all source files and commands necessary for distribution are included in a zipped file). For the purposes of this book, we utilize only SPDs for the installation of TEMAs using IBM Tivoli Configuration Manager. In your environment, you can use either SPDs or SPBs. For more information on IBM Tivoli Configuration Manager you can refer to *Certification Study Guide: IBM Tivoli Configuration Manager (ITCM) Version 4.2.2*, SG24-6691.

You can download the TEMA SPD files and silent response files and scripts from the IBM Redbooks Web site. See Appendix B, “Additional material” on page 803.

11.2 Prerequisites

The SPD provides a minimal initial configuration. The agent should at least be able to start and connect to the Tivoli Enterprise Monitoring Server, so that the configuration facility provided with IBM Tivoli Monitoring 6.1 can be used.

To develop SPDs, some additional files might be required, so a silent response file customized with SPD variables for each agent and some scripts to execute silent installation are provided. These files should reside in a local directory during the SPDs build.

Note: The SPDs have been generated using the Tivoli Software Package Editor 4.2.2. This means that to build and deploy the IBM Tivoli Monitoring Agents SPDs, IBM Tivoli Configuration Manager 4.2.2 or higher is required.

11.3 SPDs developed with this solution

The following tables list the IBM Tivoli Monitoring 6.1 Platform Agents that are available as SPDs and the respective SPD file name.

Table 11-1 Windows Platform Agents

IBM Tivoli Monitoring 6.1 Agent name	SPD name
Monitoring Agent for Windows OS	KNTWICMA.SPD
Universal Agent	KUMWICMA.SPD
Warehouse Proxy	KHDWICMA.SPD
Summarization and Pruning agent	KSYWICMA.SPD

Table 11-2 UNIX Platform Agents

IBM Tivoli Monitoring 6.1 Agent name	SPD name
Monitoring Agent for UNIX OS	UX<PLAF>[a].SPD
Monitoring Agent for UNIX Logs	UL<PLAF>.SPD
Summarization and Pruning agent[b]	SY<PLAF>.SPD
Universal Agent	UM<PLAF>.SPD

a. In this table, <PLAF> is one of the following: SOLARIS, AIX, or HP.
b. The Summarization and Pruning agent is not available for HP-UX platform.

Table 11-3   Linux and zLinux platform agents

IBM Tivoli Monitoring 6.1 Agent name	SPD name
Monitoring Agent for Linux OS	LZ<PLAF>[a].SPD
Monitoring Agent for UNIX Logs	UL<PLAF>.SPD
Summarization and Pruning agent[b]	SY<PLAF>.SPD
Universal Agent	UM<PLAF>.SPD

a. In this table, <PLAF> is one of the following: LINUX or ZLINUX.
b. The Summarization and Pruning agent is not available for Linux on OS390.

Also, Application Agent SPDs are developed as shown in the following tables.

Table 11-4   Windows Application Agents

IBM Tivoli Monitoring 6.1 Agent name	SPD name
Monitoring Agent for Microsoft SQL Server	KOQWICMA.SPD
Monitoring Agent for Sybase Server	KOYWICMA.SPD
Monitoring Agent for Oracle	KORWICMA.SPD
Monitoring Agent for DB2	KUDWICMA.SPD
Monitoring Agent for Active Directory	K3ZWICMA.SPD
Monitoring Agent for Microsoft Exchange Server	KEXWICMA.SPD
Composite Application Manager for Response Time Tracking Monitoring Agent	KT2WICMA.SPD
Composite Application Manager for WebSphere Monitoring Agent	KYNWICMA.SPD

Table 11-5   AIX/Solaris Application Agents

IBM Tivoli Monitoring 6.1 Agent name	SPD name
Monitoring Agent for DB2	UD<PLAF>[a].SPD
Monitoring Agent for Oracle	OR<PLAF>.SPD
Monitoring Agent for Sybase Server	OY<PLAF>.SPD
Composite Application Manager for Response Time Tracking Monitoring Agent	T2<PLAF>.SPD
Composite Application Manager for WebSphere Monitoring Agent	YN<PLAF>.SPD

a. In this table, <PLAF> is one of the following: SOLARIS, AIX.

Table 11-6 HP-UX Application Agents

IBM Tivoli Monitoring 6.1 Agent name	SPD name
Monitoring Agent for DB2	UDHP.SPD
Monitoring Agent for Oracle	ORHP.SPD
Composite Application Manager for Response Time Tracking Monitoring Agent	T2HP.SPD
Composite Application Manager for WebSphere Monitoring Agent	YNHP.SPD

Table 11-7 Linux and zLinux Application Agents

IBM Tivoli Monitoring 6.1 Agent name	SPD name
Composite Application Manager for Response Time Tracking Monitoring Agent	T2<PLAF>[a].SPD
Composite Application Manager for WebSphere Monitoring Agent	YN<PLAF>.SPD

a. In this table, *<PLAF>* is one of the following: LINUX or ZLINUX.

11.4 SPD structure

Each SPD contains the list of files required to install the agent and some files required for the install process, such as the silent response file used by the silent install process.

During the install operation, the SPDs deploy the required files under the CandleHome directory, then launch the silent install process available for each agent.

During the remove operation, the SPDs launch the silent uninstall process for that agent, then remove the files deployed during the install operation.

11.4.1 SPD naming convention

For Windows, each SPD filename is an acronym of the agent to install; the name used is the same as the silent.txt tag acronym for that agent. (That is, KNTWICMA.SPD is the SPD to distribute the Windows OS Agent.)

For UNIX, the name is composed of the two-letter product code and the operating system name. For example, the UNIX OS Agent for Solaris SPD filename is UXSOLARIS.SPD.

The software package name is the same as the SPD filename plus the version. Following the example above, the software package name for the Windows OS Agent is KMNTWICMA.6.1.0.

All the software packages must have Version 6.1.0, except for the Composite Application Manager Agents, as described in Table 11-8.

Table 11-8 Composite Application Manager software package names

IBM Tivoli Monitoring 6.1 Agent name	**Software package name**
Composite Application Manager for Response Time Tracking Monitoring Agent (Windows OS)	KT2WICMA.6.0.0
Composite Application Manager for WebSphere Monitoring Agent (Windows OS)	KYNWICMA.6.0.0
Composite Application Manager for Response Time Tracking Monitoring Agent (UNIX, Linux, and zLinux)	T2*<PLAF>*[a].6.0.0
Composite Application Manager for WebSphere Monitoring Agent (UNIX, Linux, and zLinux)	YN*<PLAF>*.6.0.0

a. In this table, *<PLAF>* is either SOLARIS, AIX, HP, LINUX, or ZLINUX.

Note: From the Tivoli Desktop you might change the software package name, but changing the name could lead to deployment errors. In fact, the software package name might be used to set dependencies in other agent software packages.

11.4.2 SPD variables and dependencies

The SPD contains some user-defined variables needed to build and deploy the package. These variables can be categorized into two sets:

a. Variables needed to convert the SPD in SPB that the user must pass in. These variables have a default value that could not be valid in your environment.

b. Variables used to configure the install. A default variable for them is provided, and it represents the same default variable that comes with IBM Tivoli Monitoring 6.1 Agents. Review them and perform the appropriate changes as per your configuration.

The following variables belong to the first set:

source_dir	Local directory that contains the Agent binaries
silentfiles_dir	Local directory that contains the reworked response files and scripts

The following variables belong to the second set:

CandleHome	Destination install directory. Default: /opt/IBM/ITM
CandleEncryptionKey	Encryption key. Default: IBMTivoliMonitoringEncryptionKey
Others	Variables needed for initial configuration, such as CMS server, protocol, and port number. (Refer to Table 11-9 on page 642 for a more detailed list of variables you can use).

The "save default variables" flags in each SPD will be set to FALSE, so that each package not using the default CandleHome must specify the CandleHome value provided to the OS Agent SPD.

The OS Agents use a configuration file that stores all variables belonging to the second set that are specified either in the SPD configuration or at SPB installation time. The same file is used by the other UNIX Platform Agents during the configuration phase to inherit the network configuration parameters. This is done automatically by the Windows Platform Agents that inherit the network configuration parameters from the Windows OS Agent configuration.

Note: The OS Agent package is always installed first and is a prerequisite for all other agent packages, both for platform and for application agents. Using this approach, the packages can be smaller because the great part of the install files and prerequisites will be deployed only by the OS Agent package, and all other packages leverage the files distributed and installed by the OS Agent package. To achieve this, all of the packages have a dependency to the OS Agent package.

The dependency that will be set for all the Windows Agents packages is:

```
dependency = "$(installed_software) >= KNTWICMA.6.1.0"
```

This also means that the OS Agent package cannot be removed until all other packages are removed.

11.4.3 SPD stanzas

The SPDs contain a generic container and an executable program stanza:

- Generic container

  The generic container contains directories and files that are customized for Tivoli Configuration Manager deployment, such as new silent response files and batch or shell scripts needed to install/uninstall/configure.

  Note: On Windows only, the generic container also deploys the install files needed by the agent installation (setup.exe, .cab, .msi files).

- Executable program stanza

  The executable program stanza contains:

  – The commands to stop/start the agents using the **`itmcmd start`** ***`pc`*** command, where ***`pc`*** is the product code (for example, ux for Monitoring Agent for UNIX OS).

    Note: The SPD to install the Agent contains the start command for the agent as the last command. The SPD to uninstall the Agent will contain the stop command for the agent as first command.

  – The silent install/uninstall command to execute, and on UNIX also the required install files as co-requisite files. The configuration command for the UNIX OS Agent is:

    ```
    itmcmd config -A -p <config_filename> pc
    ```

    On Windows, the command that is executed to silently install the agents is:

    ```
    cmd    /c start /wait $(CandleHome)\InstSpbs\Windows\setup.exe /w
           /z"/sf$(CandleHome)\InstSpbs\<silent_install_response_file>" /s
           /f2"$(temp_dir)\<silent_install_log_file>"
    ```

    The silent uninstall on Windows requires that the install files remain on the machine after the deployment. The files are put on the machine in:

    ```
    $(CandleHome)\InstSpbs
    ```

    It will be reused by the uninstall, and the SPB remove action will clean up the directory.

    On UNIX, the commands that are executed to silently install/uninstall the agents are, respectively:

    ```
    $(CandleHome)/InstSpbs/UNIX/install.sh -q -h $(CandleHome) -p
          $(CandleHome)/InstSpbs/<silent_install_response_file>
    ```

```
spb_uninstall.sh $(CandleHome) ux $(os_name)
```

spb_uninstall.sh is a script that checks whether the product is installed, gets the platform code required to perform a silent install, then, to perform the actual silent uninstall, launches:

```
$(CandleHome)/bin/uninstall.sh -f pc platform
```

The standard output and standard error of the UNIX commands are redirected to:

```
$(temp_dir)\<SPD_NAME>.out
```

The same file is used for the Windows commands, but the file is empty because the silent install log file is used instead.

Refer to 11.7.1, “Installation of Monitoring Agent for UNIX OS and Universal Agent on AIX” on page 643, which illustrates both SPD customization and SPB installation.

11.5 Configuring IBM Tivoli Monitoring 6.1 Agent SPDs

Before importing the SPDs in the Tivoli environment some configuration is needed.

The SPDs contains two variables that must be set before the import operation:

source_dir	Full path to the IBM Tivoli Monitoring 6.1 Agent install images
silentfiles_dir	Full path to the response and scripts files provided with this solution

The source_dir variable must be set to the directory containing the Agent images. This directory can be located on the CD-ROM containing the IBM Tivoli Monitoring images.

The silentfiles_dir variable must be set to the path containing the files that are provided with this solution. When you unzip the files provided by this solution, two directories are created on the destination directory you selected for the unzip; they must reside on the Tivoli Configuration Manager Source Host machine:

SilentFiles	Contains all response files and scripts needed to perform the silent install recalled by the SPB installation.
Spds	Contains two subdirectories, Unix and Windows. The Unix subdirectory will contain other directories for each OS type. These directories contain the SPD files for both the Platform and Application Agents, as listed in 11.3, “SPDs developed with this solution” on page 635.

11.6 Importing V6.1 Agent SPDs in a Tivoli environment

After the source_dir and silentfiles_dir directories have been set to a valid value, the SPDs can be imported using the Tivoli Desktop, as documented in *IBM Tivoli Configuration Manager User's Guide for Software Distribution v4.2.3*, SC23-4711, or using the **wimpspo** command as documented in *IBM Tivoli Configuration Manager Reference Manual for Software Distribution*, SC23-4712.

11.7 Deploying IBM Tivoli Monitoring 6.1 Agents SPDs

This section provides additional details for deploying IBM Tivoli Monitoring Agents.

Important: The OS Agent must always be deployed and installed first using IBM Tivoli Configuration Manager Software Distribution. All other Agents require that the OS Agent is already deployed and installed on the target machine.

The OS Agent has a list of variables that can be set to configure the Agent connection to the TEMS. Refer to Table 11-9 for a list of the variables that you can use.

Table 11-9 Variables can be specified to configure Agent connection to TEMS

Variable name	Variable description
CandleHome	IBM Tivoli Monitoring 6.1 Agent Installation Directory
CandleEncryptionKey	Encryption key to encrypt local passwords and IDs
TEMSHostname	Host name for the monitoring server
NeworkProtocol	Protocol to use to communicate with the monitoring server
IPPipePortNumber	IP.PIPE port number for the monitoring server
PortNumber	IP.UDP port number for the monitoring server
IPSpipePortNumber	IP.SPIPE port number for the monitoring server
SNANetName	SNA network identifier for your location
SNALUNName	LU name for the monitoring server
SNALogMode	Name of the LU6.2 LOGMODE

Variable name	Variable description
BackupNetworkProtocol	Backup protocol to use to communicate with the monitoring server
UseSecondaryTEMS	Set to YES if you are using a secondary monitoring server
SecondaryTEMSHostname	Host name for the secondary monitoring server

Note: The value of the CandleHome variable must be the same for all IBM Tivoli Monitoring 6.1 Agents deployed on the same target machine; otherwise the Agent deployment will fail.

The variables above enable you to configure the primary and backup TEMS connection; they also enable you to specify a backup network protocol that will be used for both configured TEMS.

Note: The UseSecondaryTEMS variable only accepts YES or NO as values. If YES is selected, the BackupNetworkProtocol and SecondaryTEMSHostname will be used to value the CT_CMSLIST and KDC_FAMILIES variables in the configuration file for the agent you are deploying. (If you are installing the UNIX OS Agent, the file will be $CANDLEHOME/config/ux.config).

To deploy an agent SPD, refer to *IBM Tivoli Configuration Manager User's Guide for Software Distribution*, SC23-4711.

11.7.1 Installation of Monitoring Agent for UNIX OS and Universal Agent on AIX

This section provides information about a real implementation of IBM Tivoli Monitoring, Monitoring Agent for UNIX OS, and Universal Agent on an AIX 5.3 32-bit operating system. You can find the SPD files referenced in this section in Appendix A, "TEMA Software Package Definition examples" on page 753. You can also download them from the IBM Redbooks Web site. For instructions on how to download these, refer to Appendix B, "Additional material" on page 803.

This section has several screen shots that show the configuration steps required for a successful installation of both the Monitoring Agent for UNIX OS and Universal Agent.

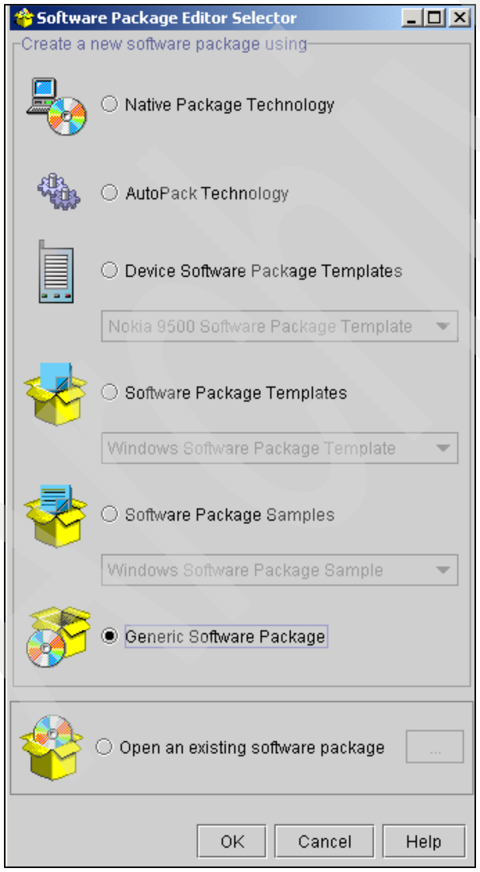
Monitoring Agent for UNIX OS

The following steps show how to install the Monitoring Agent for UNIX OS using IBM Tivoli Configuration Manager Software Distribution.

1. Download the SPD file from the Additional Material link in the IBM Redbooks Web site. For download instructions see Appendix B, “Additional material” on page 803.
2. Use the IBM Tivoli Software Package Editor V4.2.2 or higher to import the SPD file. For the Monitoring Agent for UNIX OS, the SPD file name is UXAIX.SPD as shown in Table 11-2 on page 635.

   The following figures guide you through the SPD import process performed using the IBM Tivoli Software Package Editor for Endpoint V4.2.2:

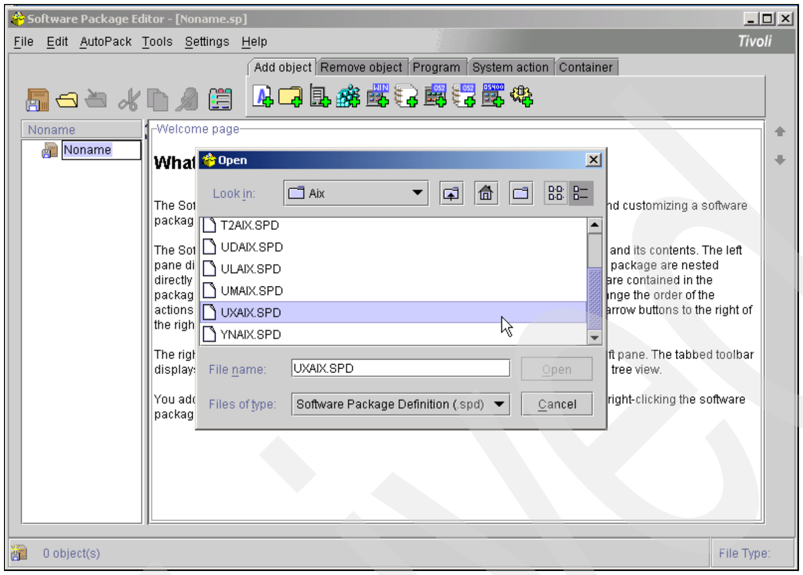
   a. After the IBM Tivoli Software Package Editor V4.2.2 is deployed, double-click its icon on the desktop. The window in Figure 11-1 appears.


Software Package Editor Selector
Create a new software package using
Native Package Technology
AutoPack Technology
Device Software Package Templates
Nokia 9500 Software Package Template
Software Package Templates
Windows Software Package Template
Software Package Samples
Windows Software Package Sample
Generic Software Package
Open an existing software package
...
OK
Cancel
Help

Selected: ● Generic Software Package

Figure 11-1 IBM Tivoli Software Package Editor launch dialog

b. The Generic Software Package selection presents an empty container. Select **File** → **Open** and locate the SPD file called UXAIX.SPD.


Software Package Editor - [Noname.sp]

Open

Look in: Aix

T2AIX.SPD
UDAIX.SPD
ULAIX.SPD
UMAIX.SPD
UXAIX.SPD
YNAIX.SPD

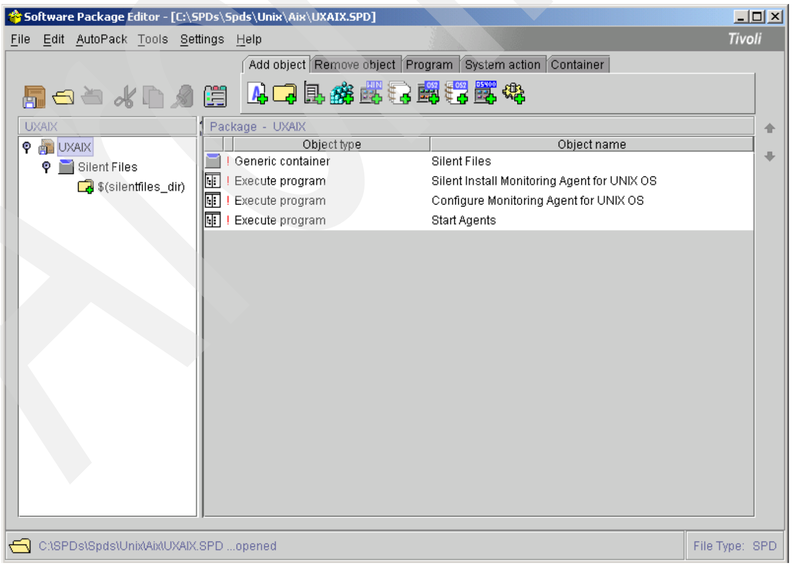
File name: UXAIX.SPD

Files of type: Software Package Definition (.spd)

0 object(s)

Figure 11-2 IBM Tivoli Software Package Editor UMAIX.SPD selection

c. When the import operation completes, the window in Figure 11-3 appears.


Software Package Editor - [C:\SPDs\Spds\Unix\Aix\UXAIX.SPD]

UXAIX
Silent Files
$(silentfiles_dir)

Package - UXAIX

Object type	Object name
! Generic container	Silent Files
! Execute program	Silent Install Monitoring Agent for UNIX OS
! Execute program	Configure Monitoring Agent for UNIX OS
! Execute program	Start Agents

C:\SPDs\Spds\Unix\Aix\UXAIX.SPD ...opened

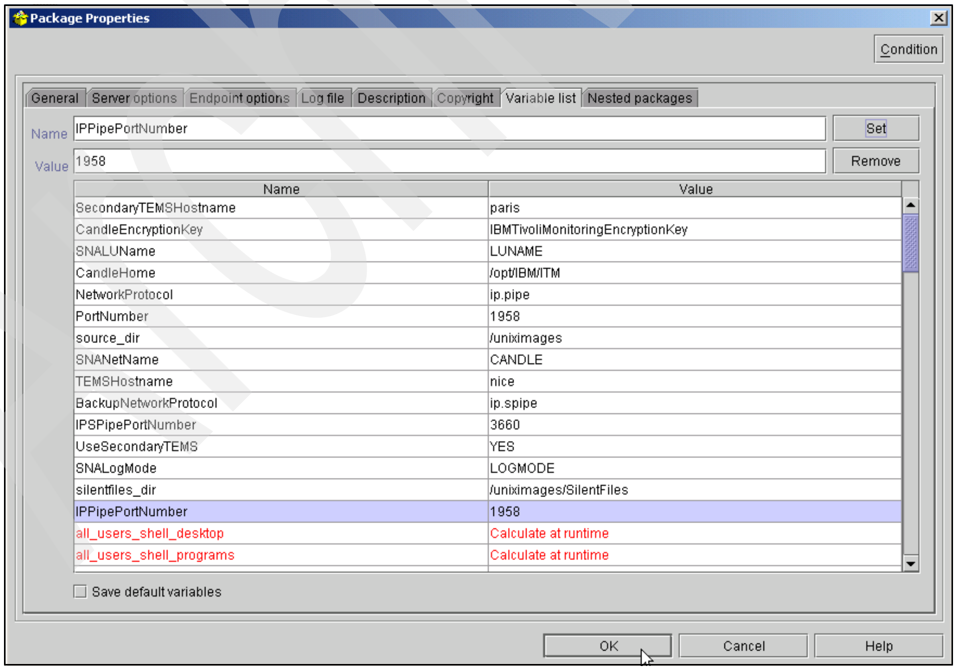
File Type: SPD

Figure 11-3 IBM Tivoli Software Package Editor UMAIX.SPD view

3. Change all the required variables that apply to your environment:
   – souce_dir and silentfiles_dir are required for a successful SPD build on the IBM Tivoli Software Distribution Source Host machine and must point to the IBM Tivoli Monitoring 6.1 image for the operating system you are installing (for example, the path used to **`untar`** the UNIX image). Optionally, these directories can reside on the IBM Tivoli Software Distribution Source Host machine.
   – CandleHome is defaulted to opt/IBM/ITM for AIX.
   – TEMSHostname is set to the machine host name (the one you are installing the Agent to) but in most cases, this must be changed to match the TEMS host name.
   – IPPortNumber, PortNumber, and IPSpipePortNumber refer respectively to IP.PIPE, IP.UDP, and IP.SPIPE communication protocol, and the default for them is 1918 for IP.PIPE and IP.UDP and 3660 for IP.SPIPE. In this example, we have used port 1958.

   Other IBM Tivoli Monitoring 6.1 configuration variables must be changed based on your configuration. Refer to Table 11-9 on page 642 for a detailed list and explanation of the other variables.

   Figure 11-4 shows the variables configuration dialog for the UXAIX.SPD.



Name: IPPipePortNumber

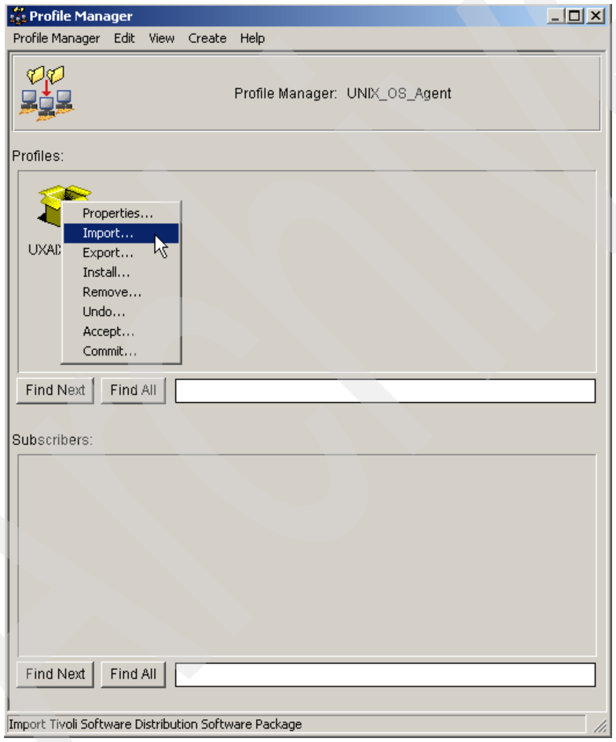
Value: 1958

Name	Value
SecondaryTEMSHostname	paris
CandleEncryptionKey	IBMTivoliMonitoringEncryptionKey
SNALUName	LUNAME
CandleHome	/opt/IBM/ITM
NetworkProtocol	ip.pipe
PortNumber	1958
source_dir	/uniximages
SNANetName	CANDLE
TEMSHostname	nice
BackupNetworkProtocol	ip.spipe
IPSPipePortNumber	3660
UseSecondaryTEMS	YES
SNALogMode	LOGMODE
silentfiles_dir	/uniximages/SilentFiles
IPPipePortNumber	1958
all_users_shell_desktop	Calculate at runtime
all_users_shell_programs	Calculate at runtime

Save default variables

Figure 11-4 UXAIX.SPD variable list

4. Save the modifications made to the SPD file.
5. Configure the IBM Tivoli Management Framework resources to contain the SPB file as follows:
   a. Create a Policy Region.
   b. Assign ProfileManager and SoftwarePackage as Managed Resources for the Policy Region.
   c. Create a Profile Manager.
   d. Create an empty Software Package. For this example, the name must be UXAIX.6.1.0. Refer to 11.4.1, “SPD naming convention” on page 637 for details about the SPB naming convention.
   e. Import the SPD file into the empty software package that was created in the previous step, as shown in Figure 11-5.



Profile Manager: UNIX_OS_Agent

Import Tivoli Software Distribution Software Package

Figure 11-5 Import SPD dialog

6. Select the endpoint with the IBM Tivoli Software Package Editor used to modify the SPD file as shown in Figure 11-6.

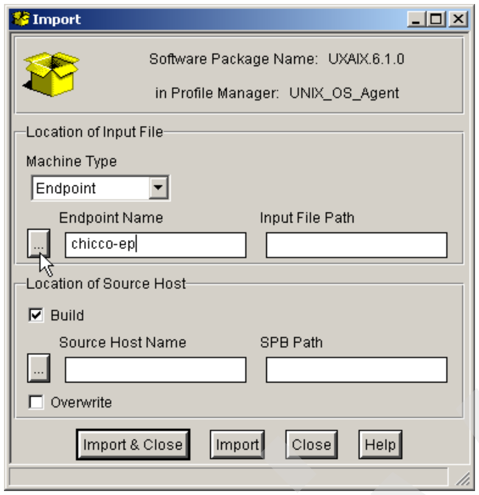
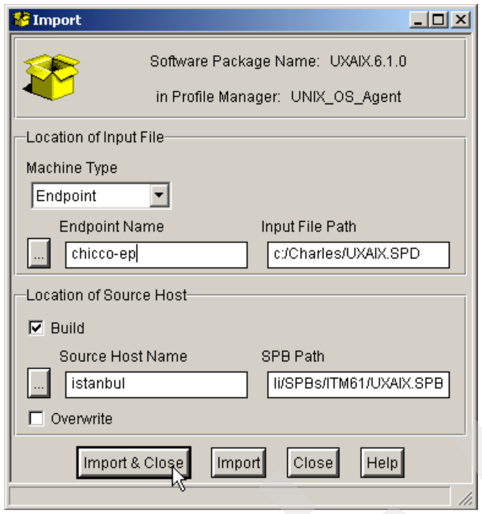


Figure 11-6 Select Endpoint from Software Package import dialog

Tip: If the source is an endpoint, you cannot browse the list of endpoints; enter the endpoint label in the Machine Type field and click the browse (**...**) button; at this point you can browse the endpoint disk drives.

7. Provide the source host information to build the SPB as shown in Figure 11-7.


Import

Software Package Name: UXAIX.6.1.0

in Profile Manager: UNIX_OS_Agent

Location of Input File

Machine Type: Endpoint

Endpoint Name: chicco-ep

Input File Path: c:/Charles/UXAIX.SPD

Location of Source Host

☑ Build

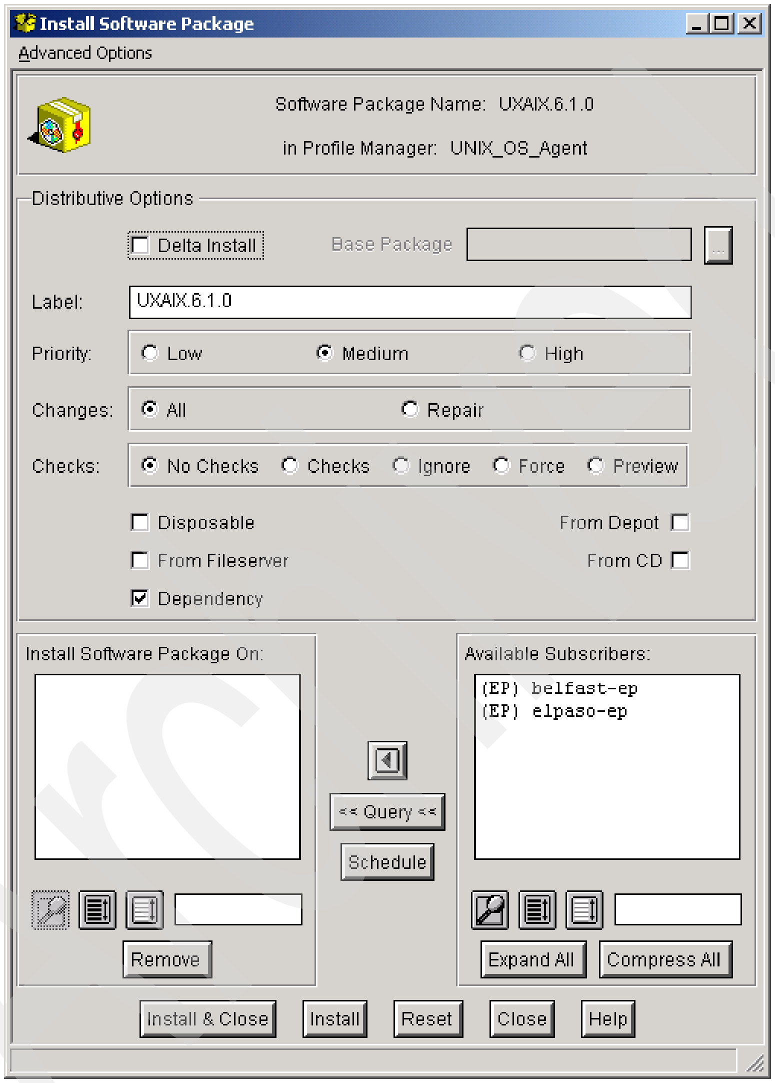
Source Host Name: istanbul

SPB Path: li/SPBs/ITM61/UXAIX.SPB

☐ Overwrite

Figure 11-7 IBM Tivoli Software Distribution Source Host information

8. Now that the SPB has been built, you can install the package. As for the import operation, right-click on the built Software Package icon and select **Install**. Figure 11-8 shows the install dialog that will be displayed.



Software Package Name: UXAIX.6.1.0

in Profile Manager: UNIX_OS_Agent

Distributive Options

Delta Install [ ] Base Package

Label: UXAIX.6.1.0

Priority: Low [✓ Medium] High

Changes: [✓ All] Repair

Checks: [✓ No Checks] Checks Ignore Force Preview

[ ] Disposable From Depot [ ]

[ ] From Fileserver From CD [ ]

[✓] Dependency

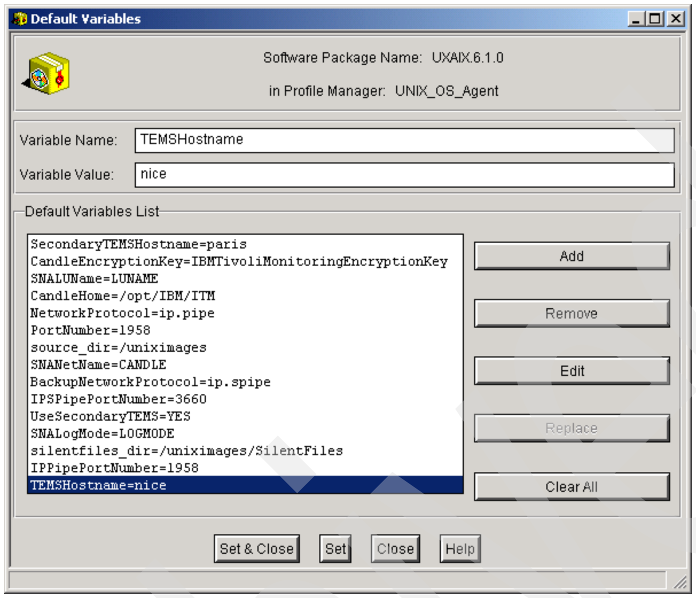
Install Software Package On:

Available Subscribers:

```
(EP) belfast-ep
(EP) elpaso-ep
```

Figure 11-8 UMAIX.6.1.0 Software Package install dialog

9. You can review and eventually change the **Default Variables** set for the SPD from the **Advanced Options** menu item (Figure 11-9).



Software Package Name: UXAIX.6.1.0

in Profile Manager: UNIX_OS_Agent

Variable Name: TEMSHostname

Variable Value: nice

Default Variables List

```
SecondaryTEMSHostname=paris
CandleEncryptionKey=IBMTivoliMonitoringEncryptionKey
SNALUName=LUNAME
CandleHome=/opt/IBM/ITM
NetworkProtocol=ip.pipe
PortNumber=1958
source_dir=/uniximages
SNANetName=CANDLE
BackupNetworkProtocol=ip.spipe
IPSPipePortNumber=3660
UseSecondaryTEMS=YES
SNALogMode=LOGMODE
silentfiles_dir=/uniximages/SilentFiles
IPPipePortNumber=1958
TEMSHostname=nice
```

Figure 11-9 Set Default Variables dialog for UNIX OS

10. You are now ready to install the Software Package. Select the target endpoint and click **Install** or **Install & Close**.

11. You can monitor the distribution as shown in Example 11-1.

Example 11-1 Software Package install progress for UNIX OS

```
You can use the wmdist command line interface or the Distribution Status
console from the Tivoli Desktop to check the distribution progress:

[root@istanbul][/]-> wmdist -la
Name              Distribution ID     Targets Completed  Successful      Failed
UXAIX.6.1.0    (install) 1567175343.239          1     0(  0%)     0(  0%)
0(  0%)

The distribution ID is 239 (you can simply use the last number after the dot)

[root@istanbul][/Tivoli/SPBs/ITM61]-> wmdist -I istanbul

Repeater: istanbul

Jobs in SEND queue:     1
Jobs in RECEIVE queue:  0
```

```
        === Session Information ===

Low:    available = 40         used = 0
Medium: available = 9          used = 1
High:   available = 5          used = 0

        === Distribution Information ===

External Id:    1567175343.239
Internal Id:    1567175343.239
Label:          UXAIX.6.1.0    (install)
Priority:       3
Application:    Software Distribution

Target: belfast-ep      State: RECEIVING 2% (5292032/182026416)

Whit the above command, you can see that the actual Software Package is being
distributed to the Endpoint named belfast-ep.

[root@istanbul][/Tivoli/SPBs/ITM61]-> wmdist -I istanbul

Repeater: istanbul

Jobs in SEND queue:     1
Jobs in RECEIVE queue:  0

        === Session Information ===

Low:    available = 40         used = 0
Medium: available = 9          used = 1
High:   available = 5          used = 0

        === Distribution Information ===

External Id:    1567175343.239
Internal Id:    1567175343.239
Label:          UXAIX.6.1.0    (install)
Priority:       3
Application:    Software Distribution

Target: belfast-ep      State: RECEIVING 62% (113852416/182026416)

[root@istanbul][/Tivoli/SPBs/ITM61]-> wmdist -I istanbul

Repeater: istanbul

Jobs in SEND queue:     1
Jobs in RECEIVE queue:  0
```

```
        === Session Information ===

Low:    available = 40         used = 0
Medium: available = 9          used = 1
High:   available = 5          used = 0

        === Distribution Information ===

External Id:    1567175343.239
Internal Id:    1567175343.239
Label:          UXAIX.6.1.0    (install)
Priority:       3
Application:    Software Distribution

Target: belfast-ep      State: RECEIVING 100% (182026416/182026416)

Now that the receive operaion completed, the actual installation is in progress
on the Endpoint side and you will see the RECEIVING 100% stauts unless the
local installation completes, and the distribution result is sent back to the
Tivoli Management Region server.

Once the distribution report has been sent by the Endpoint, the above command
does not show the Distribution Information anymore. You can check the
distribution result using the wmdist -e dist_id command.

[root@istanbul][/Tivoli/SPBs/ITM61]-> wmdist -I istanbul

Repeater: istanbul

Jobs in SEND queue:     0
Jobs in RECEIVE queue:  0

        === Session Information ===

Low:    available = 40         used = 0
Medium: available = 10         used = 0
High:   available = 5          used = 0

[root@istanbul][/Tivoli/SPBs/ITM61]-> wmdist -e 239
istanbul              SUCCESSFUL     2005.11.14 08:19:12  2005.11.14 08:32:26
belfast-ep            SUCCESSFUL     2005.11.14 08:19:13  2005.11.14 08:32:26

From to the Endpoint, you can check the status of the installation with the
cinfo command:

[root@belfast][/]-> cinfo -r

*********** Mon Nov 14 13:20:59 CST 2005 ******************
```

```
User      : root         Group: system bin sys security cron audit lp
Host name : belfast      Installer Lvl: 400 / 100
CandleHome: /opt/IBM/ITM
***********************************************************
Host     Prod    PID     Owner   Start    ID  ..Status
belfast  ux      20642   root    09:46:50 None ..running
```

This command lists the running processes.

```
[root@belfast][/]-> cinfo -i

*********** Mon Nov 14 13:21:03 CST 2005 *****************
User      : root         Group: system bin sys security cron audit lp
Host name : belfast      Installer Lvl: 400 / 100
CandleHome: /opt/IBM/ITM
***********************************************************
...Product inventory

ax      IBM Tivoli Monitoring Shared Libraries
         aix513  Version: 06.10.00.00

jr      Tivoli Enterprise-supplied JRE
         aix513  Version: 400 Rel: 100

ui      Tivoli Enterprise Services User Interface
         aix513  Version: 06.10.00.00

ux      Monitoring Agent for UNIX OS
         aix513  Version: 06.10.00.00
```

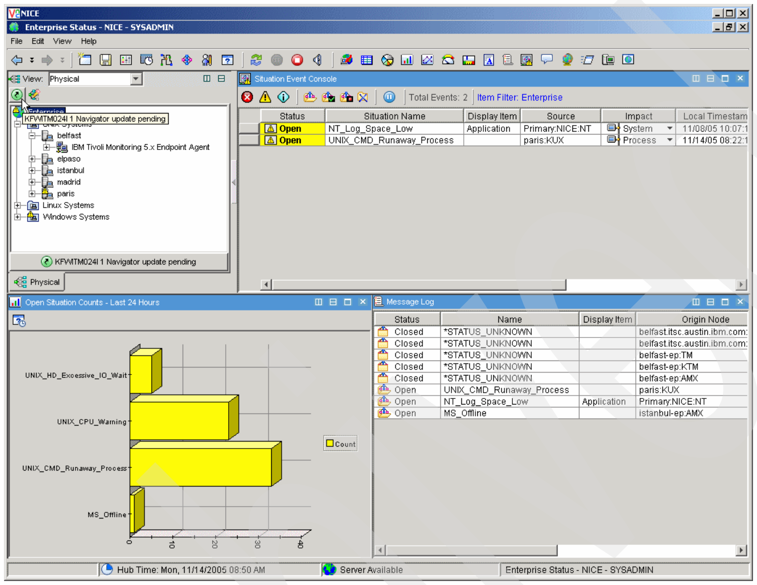
This command lists the IBM Tivoli Monitoring 6.1 Product inventory installed on the Endpoint.

You can also check the Agent configuration running the below command:

```
[root@belfast][/opt/IBM/ITM/config]-> itmcmd config -A -g ux
Agent configuration started...
#-> Configuration settings. Product=ux Key=aix513
ARCHITECTURE=tmaitm6/aix513
MIRROR=paris
HSPORTNUMBER=1958
CMSCONNECT=YES
HSIPPIPEPORTNUMBER=1958
BK2HSNETWORKPROTOCOL=none
PRIMARYIP=none
BK1HSNETWORKPROTOCOL=ip.spipe
IPPIPEPORTNUMBER=1958
IPSPIPEPORTNUMBER=3660
NETNAME=CANDLE
```

```
BK1NETWORKPROTOCOL=ip.spipe
BINARCH=aix513
HSLUNAME=LUNAME
HOSTNAME=nice
HSIPSPIPEPORTNUMBER=3660
FTO=YES
HSNETWORKPROTOCOL=ip.pipe
COMMENT=#
CANDLEHOME=/opt/IBM/ITM
BK2NETWORKPROTOCOL=none
RUNNINGHOSTNAME=belfast
LOGMODE=LOGMODE
NETWORKPROTOCOL=ip.pipe
PRODUCTCODE=ux
PORTNUMBER=1958
HSLOGMODE=LOGMODE
LUNAME=LUNAME
KDC_PARTITIONNAME=none
CELLNAME=CANDLE
JAVAHOME=/opt/IBM/ITM/JRE/aix513
HSNETNAME=CANDLE
```

12. You can open the TEP Client to confirm that the OS Agent has been installed successfully and connected to the TEMS specified by the TEMSHostname variable. Before the UNIX OS tree is created for the Agent in the TEP, you are asked to click the green icon in the top-left of the navigator workspace to apply the pending update change as shown in Figure 11-10.

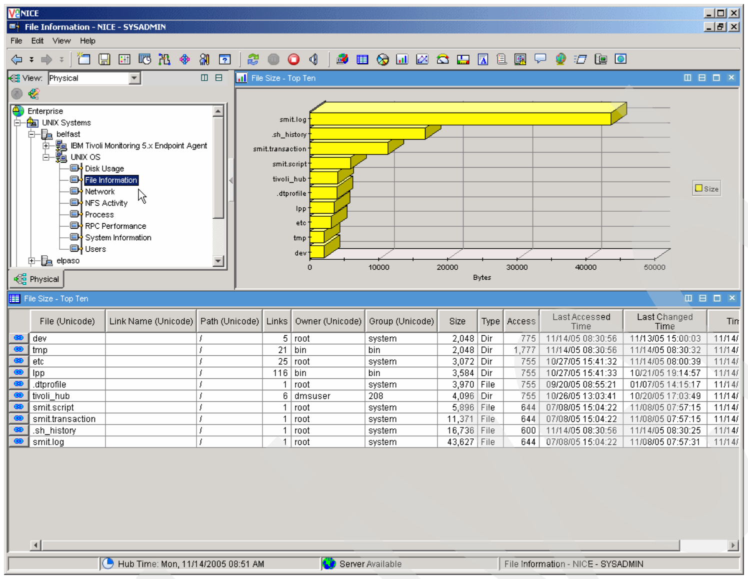

NICE
Enterprise Status - NICE - SYSADMIN
File Edit View Help
View: Physical
KFWITM024I 1 Navigator update pending
Enterprise
UNIX Systems
belfast
IBM Tivoli Monitoring 5.x Endpoint Agent
elpaso
istanbul
madrid
paris
Linux Systems
Windows Systems
KFWITM024I 1 Navigator update pending
Physical
Situation Event Console
Total Events: 2
Item Filter: Enterprise
Open Situation Counts - Last 24 Hours
UNIX_HD_Excessive_IO_Wait
UNIX_CPU_Warning
UNIX_CMD_Runaway_Process
MS_Offline
0
10
20
30
40
Count
Message Log
Hub Time: Mon, 11/14/2005 08:50 AM
Server Available
Enterprise Status - NICE - SYSADMIN

Status	Situation Name	Display Item	Source	Impact	Local Timestam
Open	NT_Log_Space_Low	Application	Primary:NICE:NT	System	11/08/05 10:07:1
Open	UNIX_CMD_Runaway_Process		paris:KUX	Process	11/14/05 08:22:1

Status	Name	Display Item	Origin Node
Closed	*STATUS_UNKNOWN		belfast.itsc.austin.ibm.com:
Closed	*STATUS_UNKNOWN		belfast.itsc.austin.ibm.com:
Closed	*STATUS_UNKNOWN		belfast-ep:TM
Closed	*STATUS_UNKNOWN		belfast-ep:KTM
Closed	*STATUS_UNKNOWN		belfast-ep:AMX
Open	UNIX_CMD_Runaway_Process		paris:KUX
Open	NT_Log_Space_Low	Application	Primary:NICE:NT
Open	MS_Offline		istanbul-ep:AMX

Figure 11-10 UNIX OS Agent TEP Navigator update pending

13. You can now navigate through the Tivoli Enterprise Portal views as shown in Figure 11-11.



File Size - Top Ten

File (Unicode)	Link Name (Unicode)	Path (Unicode)	Links	Owner (Unicode)	Group (Unicode)	Size	Type	Access	Last Accessed Time	Last Changed Time	Tin
dev		/	5	root	system	2,048	Dir	775	11/14/05 08:30:56	11/13/05 15:00:03	11/14/
tmp		/	21	bin	bin	2,048	Dir	1,777	11/14/05 08:30:56	11/14/05 08:30:32	11/14/
etc		/	25	root	system	3,072	Dir	755	10/27/05 15:41:32	11/14/05 08:00:39	11/14/
lpp		/	116	bin	bin	3,584	Dir	755	10/27/05 15:41:33	10/21/05 19:14:57	11/14/
.dtprofile		/	1	root	system	3,970	File	755	09/20/05 08:55:21	01/07/05 14:15:17	11/14/
tivoli_hub		/	6	dmsuser	208	4,096	Dir	755	10/26/05 13:03:41	10/20/05 17:03:49	11/14/
smit.script		/	1	root	system	5,896	File	644	07/08/05 15:04:22	11/08/05 07:57:15	11/14/
smit.transaction		/	1	root	system	11,371	File	644	07/08/05 15:04:22	11/08/05 07:57:15	11/14/
.sh_history		/	1	root	system	16,736	File	600	11/14/05 08:30:56	11/14/05 08:30:25	11/14/
smit.log		/	1	root	system	43,627	File	644	07/08/05 15:04:22	11/08/05 07:57:31	11/14/

Figure 11-11 UXAIX Tivoli Enterprise Portal File Information

Universal Agent

The Universal Agent installation is very similar to the UNIX OS Agent installation. We simply repeat the steps given in “Monitoring Agent for UNIX OS” on page 644 and provide figures for the dialogs that are different from the UNIX OS Agent installation.

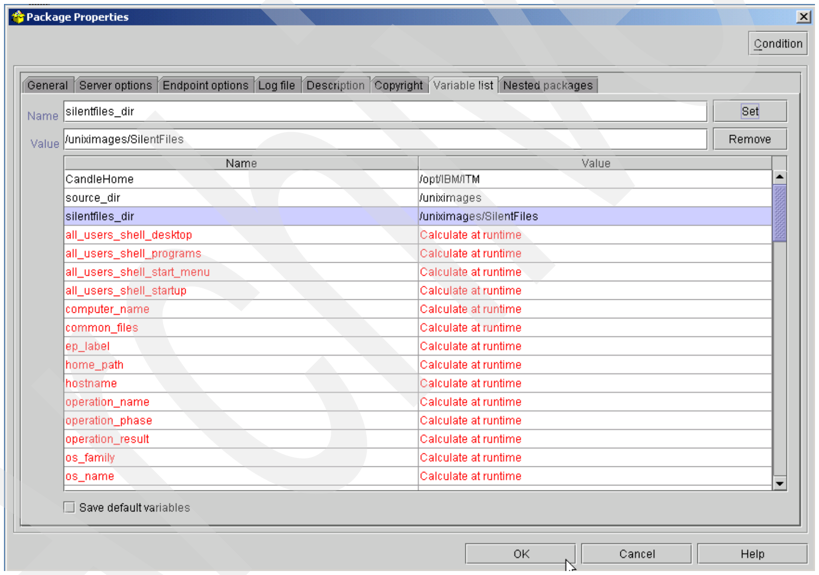
1. Download the SPD file from the Additional Material link in the IBM Redbooks Web site. For download instructions, refer to Appendix B, “Additional material” on page 803.
2. Use the IBM Tivoli Software Package Editor V4.2.2 or higher to import the SPD file. In this case, the SPD file is UMAIX.SPD as detailed in Table 11-2 on page 635.
3. Change all of the required variables that apply to your environment:
   – souce_dir and silentfiles_dir are required for a successful SPD build on the IBM Tivoli Software Distribution Source Host machine and must point to the IBM Tivoli Monitoring 6.1 image for the OS you are installing, such as the path used to **`untar`** the UNIX image. These directories can optionally reside on the IBM Tivoli Software Distribution Source Host machine.
   – CandleHome is defaulted to opt/IBM/ITM for AIX.

Note: These are the only variables pre-added to the Universal Agent SPD file.

All other variables needed to successfully connect the Universal Agent to the TEMS, such as TEMSHostname, NetworkProtocol IPPipePortNumber, and the backup TEMS path, are contained in a file created during the UNIX OS Agent installation, used by the configuration phase of the Universal Agent.

All network configuration parameters are inherited by the UNIX OS Agent, so if you want to specify a different network protocol order for the Universal Agent, after the installation run the Universal Agent configuration again and restart the service.

Figure 11-12 shows the variables configuration dialog for the UMAIX.SPD.



Name: silentfiles_dir

Value: /uniximages/SilentFiles

Name	Value
CandleHome	/opt/IBM/ITM
source_dir	/uniximages
silentfiles_dir	/uniximages/SilentFiles
all_users_shell_desktop	Calculate at runtime
all_users_shell_programs	Calculate at runtime
all_users_shell_start_menu	Calculate at runtime
all_users_shell_startup	Calculate at runtime
computer_name	Calculate at runtime
common_files	Calculate at runtime
ep_label	Calculate at runtime
home_path	Calculate at runtime
hostname	Calculate at runtime
operation_name	Calculate at runtime
operation_phase	Calculate at runtime
operation_result	Calculate at runtime
os_family	Calculate at runtime
os_name	Calculate at runtime

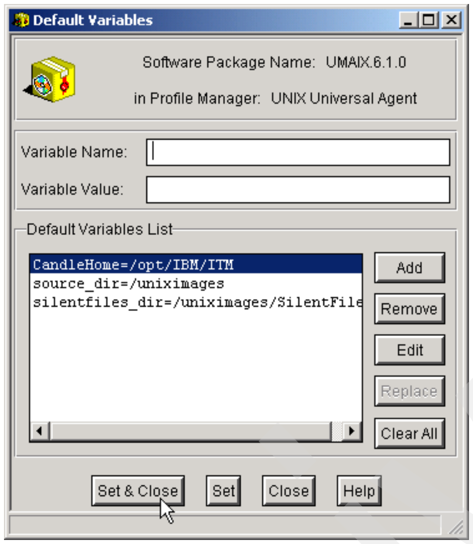
☐ Save default variables

Figure 11-12 UMAIX.SPD variable list

Refer to Table 11-9 on page 642 for a detailed list and explanation of the other variables.

4. Save the modifications made to the SPD file.
5. Configure the IBM Tivoli Management Framework resources to contain the SPB file as described in the Monitoring Agent for UNIX OS section.
6. Build the SPB.

7. Select **Install** from the right-click menu on the built Software Package icon.
8. Change the required variables, if not done at SPD level.



```
CandleHome=/opt/IBM/ITM
source_dir=/uniximages
silentfiles_dir=/uniximages/SilentFile
```

Figure 11-13 Set Default Variables dialog for Universal Agent

9. You are now ready to install the software package. Select the target endpoint and click **Install** or **Install & Close**.
10. You can monitor the distribution as shown in Example 11-2.

Example 11-2 Software package install progress for Universal Agent

```
You can use the wmdist command line interface or the Distribution Status
console from the Tivoli Desktop to check the distribution progress:

[root@istanbul][/Tivoli/usr/local/Tivoli/bin/swdis/work]-> wmdist -I istanbul

Repeater: istanbul

Jobs in SEND queue:     1
Jobs in RECEIVE queue:  0

        === Session Information ===

Low:    available = 40          used = 0
Medium: available = 9           used = 1
High:   available = 5           used = 0

        === Distribution Information ===
External Id:    1567175343.240
Internal Id:    1567175343.240
Label:          UMAIX.6.1.0    (install)
Priority:       3
Application:    Software Distribution

Target: belfast-ep       State: RECEIVING 1% (49152/2832410)

[root@istanbul][/Tivoli/usr/local/Tivoli/bin/swdis/work]-> wmdist -I istanbul

Repeater: istanbul

Jobs in SEND queue:     1
Jobs in RECEIVE queue:  0

        === Session Information ===

Low:    available = 40         used = 0
Medium: available = 9          used = 1
High:   available = 5          used = 0

        === Distribution Information ===

External Id:    1567175343.240
Internal Id:    1567175343.240
Label:          UMAIX.6.1.0    (install)
Priority:       3
Application:    Software Distribution

Target: belfast-ep       State: RECEIVING 100% (2832410/2832410)
```

Now that the receive operaion completed, the actual installation is in progress on the Endpoint side and you will see the RECEIVING 100% stauts unless the local installation completes, and the distribution result is sent back to the Tivoli Management Region server.

Once the distribution report has been sent by the Endpoint, the above command does not show the Distribution Information anymore. You can check the distribution result using the **wmdist -e *dist_id*** command.

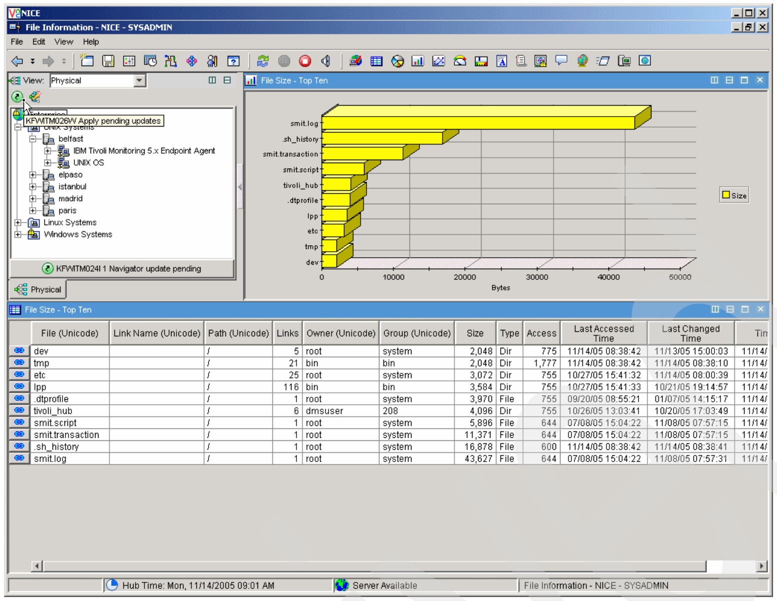
```
[root@istanbul][/Tivoli/usr/local/Tivoli/bin/swdis/work]->wmdist -e 240
istanbul              SUCCESSFUL     2005.11.14 09:05:52  2005.11.14 09:07:49
belfast-ep            SUCCESSFUL     2005.11.14 09:05:52  2005.11.14 09:07:49
```

From to the Endpoint, you can check the status of the installation with the **cinfo** command:

```
[root@belfast][/]-> cinfo -r
```

```
*********** Mon Nov 14 13:20:59 CST 2005 ******************
User      : root         Group: system bin sys security cron audit lp
Host name : belfast      Installer Lvl: 400 / 100
CandleHome: /opt/IBM/ITM
***********************************************************
Host     Prod    PID     Owner   Start    ID  ..Status
belfast  ux      20642   root    09:46:50 None ..running
belfast  um      5386    root    13:06:26 None ..running

This command lists the running processes.

[root@belfast][/]-> cinfo -i

*********** Mon Nov 14 13:21:03 CST 2005 ******************
User      : root         Group: system bin sys security cron audit lp
Host name : belfast      Installer Lvl: 400 / 100
CandleHome: /opt/IBM/ITM
***********************************************************
...Product inventory

ax      IBM Tivoli Monitoring Shared Libraries
         aix513  Version: 06.10.00.00

jr      Tivoli Enterprise-supplied JRE
         aix513  Version: 400 Rel: 100

uf      Universal Agent Framework
         aix513  Version: 06.10.00.00

ui      Tivoli Enterprise Services User Interface
         aix513  Version: 06.10.00.00

um      Universal Agent
         aix513  Version: 06.10.00.00

ux      Monitoring Agent for UNIX OS
         aix513  Version: 06.10.00.00

This command lists the IBM Tivoli Monitoring 6.1 Product inventory installed on
the Endpoint.
```

11. Open the TEP Client to confirm that the OS agent has been installed and successfully connected to the TEMS specified by the TEMSHostname variable.
12. Before the UNIX OS tree is created for the agent in the TEP, you are asked to click the green icon in the left top of the navigator workspace to apply the pending update change. (Figure 11-14 on page 662).

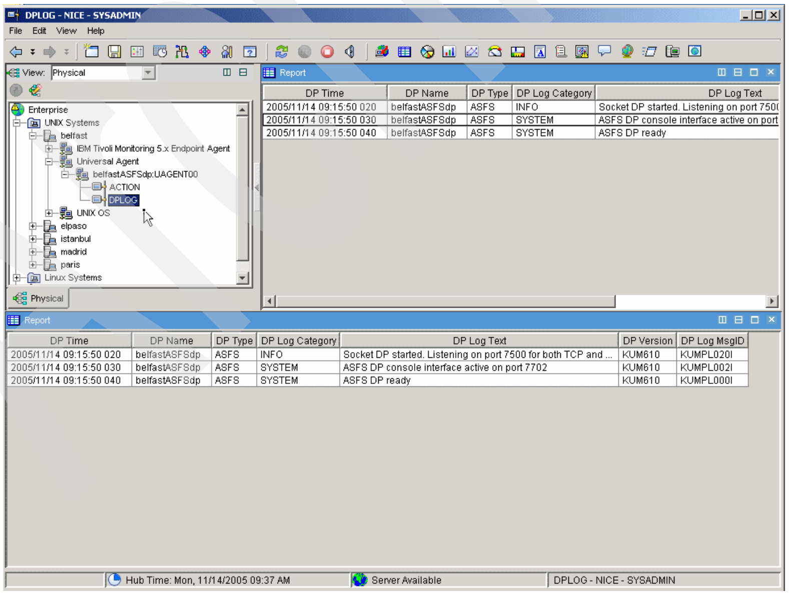


File Size - Top Ten

	File (Unicode)	Link Name (Unicode)	Path (Unicode)	Links	Owner (Unicode)	Group (Unicode)	Size	Type	Access	Last Accessed Time	Last Changed Time	Tin
	dev		/	5	root	system	2,048	Dir	775	11/14/05 08:38:42	11/13/05 15:00:03	11/14/
	tmp		/	21	bin	bin	2,048	Dir	1,777	11/14/05 08:38:42	11/14/05 08:38:10	11/14/
	etc		/	25	root	system	3,072	Dir	755	10/27/05 15:41:32	11/14/05 08:00:39	11/14/
	lpp		/	116	bin	bin	3,584	Dir	755	10/27/05 15:41:33	10/21/05 19:14:57	11/14/
	.dtprofile		/	1	root	system	3,970	File	755	09/20/05 08:55:21	01/07/05 14:15:17	11/14/
	tivoli_hub		/	6	dmsuser	208	4,096	Dir	755	10/26/05 13:03:41	10/20/05 17:03:49	11/14/
	smit.script		/	1	root	system	5,896	File	644	07/08/05 15:04:22	11/08/05 07:57:15	11/14/
	smit.transaction		/	1	root	system	11,371	File	644	07/08/05 15:04:22	11/08/05 07:57:15	11/14/
	.sh_history		/	1	root	system	16,878	File	600	11/14/05 08:38:42	11/14/05 08:38:41	11/14/
	smit.log		/	1	root	system	43,627	File	644	07/08/05 15:04:22	11/08/05 07:57:31	11/14/

Figure 11-14 Navigator updates pending icon

13. This creates the Universal Agent tree in the navigator workspace, as shown in Figure 11-15.



Report

DP Time	DP Name	DP Type	DP Log Category	DP Log Text
2005/11/14 09:15:50 020	belfastASFSdp	ASFS	INFO	Socket DP started. Listening on port 750(
2005/11/14 09:15:50 030	belfastASFSdp	ASFS	SYSTEM	ASFS DP console interface active on port
2005/11/14 09:15:50 040	belfastASFSdp	ASFS	SYSTEM	ASFS DP ready

Report

DP Time	DP Name	DP Type	DP Log Category	DP Log Text	DP Version	DP Log MsgID
2005/11/14 09:15:50 020	belfastASFSdp	ASFS	INFO	Socket DP started. Listening on port 7500 for both TCP and ...	KUM610	KUMPL020I
2005/11/14 09:15:50 030	belfastASFSdp	ASFS	SYSTEM	ASFS DP console interface active on port 7702	KUM610	KUMPL002I
2005/11/14 09:15:50 040	belfastASFSdp	ASFS	SYSTEM	ASFS DP ready	KUM610	KUMPL000I

Figure 11-15 Universal Agent TEP View

11.7.2 Removing IBM Tivoli Monitoring Agent for UNIX OS and Universal Agent on AIX

This section covers use of IBM Tivoli Configuration Manager Software Distribution to remove the Monitoring Agent for UNIX OS and Universal Agent on an AIX 5.3 32-bit operating system.

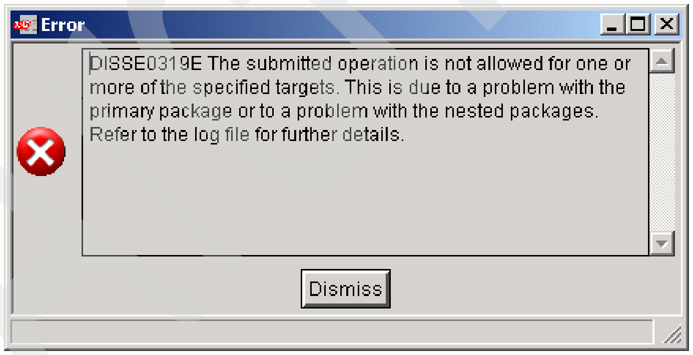
Each software package created using IBM Tivoli Configuration Manager Software Distribution can be removed easily. The remove operation executes all actions that are specified in the remove section of the SPD file that is later built (SPB) and installed on an endpoint.

This section includes screen shots that show the steps required for a successful uninstall of both the Monitoring Agent for UNIX OS and Universal Agent.

Universal Agent

The Monitoring Agent for UNIX OS is the first that we installed in the scenario described in 11.7.1, "Installation of Monitoring Agent for UNIX OS and Universal Agent on AIX" on page 643. A dependency is created for all of the other Platform and Application Agents; this means that the Monitoring Agent for UNIX OS is the last software package that can be removed.

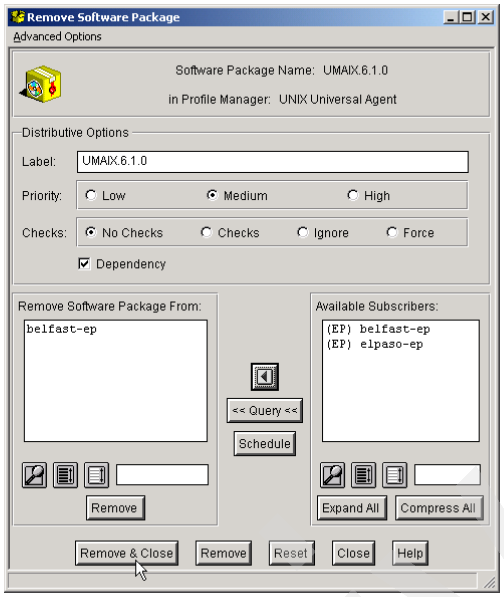
If you try to remove the Monitoring Agent for UNIX OS before any other Platform or Application Agents software package, an error message appears (Figure 11-16).



DISSE0319E The submitted operation is not allowed for one or more of the specified targets. This is due to a problem with the primary package or to a problem with the nested packages. Refer to the log file for further details.

Figure 11-16 UXAIX Software Package Remove error message

To successfully remove the Universal Agent from your endpoint, right-click the UMAIX.6.1.0 software package icon in your Profile Manager and select **Remove**, which opens the window shown in Figure 11-17 on page 664.


Remove Software Package

Advanced Options

Software Package Name: UMAIX.6.1.0

in Profile Manager: UNIX Universal Agent

Distributive Options

Label: UMAIX.6.1.0

Priority: Low [✓ Medium] High

Checks: [✓ No Checks] Checks Ignore Force

[✓ Dependency]

Remove Software Package From:

```
belfast-ep
```

Available Subscribers:

```
(EP) belfast-ep
(EP) elpaso-ep
```

Figure 11-17 UMAIX software package remove

You can monitor the distribution as shown in Example 11-3.

Example 11-3 Software Package Remove progress for Universal Agent

```
You can use the wmdist command line interface of the Distribution Status
console from the Tivoli Desktop to check the distribution progress:

root@istanbul][/]-> wmdist -la
Name              Distribution ID     Targets Completed  Successful      Failed
UMAIX.6.1.0     (remove) 1567175343.241          1      0(  0%)      0(  0%)
0(  0%)

[root@istanbul][/]-> wmdist -I istanbul

Repeater: istanbul

Jobs in SEND queue:     1
Jobs in RECEIVE queue:  0

        === Session Information ===

Low:    available = 40          used = 0
Medium: available = 9           used = 1
High:   available = 5           used = 0

        === Distribution Information ===
External Id:    1567175343.241
Internal Id:    1567175343.241
Label:          UMAIX.6.1.0    (remove)
Priority:       3
Application:    Software Distribution

Target: belfast-ep      State: RECEIVING 0% (0/0)
```

During the remove operation, the state of the distribution will remain RECEIVING 0% (0/0) unless the operation completes.

Once the distribution report has been sent by the Endpoint, the above command does not show the Distribution Information anymore. You can check the distribution result using the **wmdist -e *dist_id*** command.

```
[root@istanbul][/]-> wmdist -e 241
Name                  Status        Start Time           End Time
istanbul              SUCCESSFUL    2005.11.14 09:12:54  2005.11.14 09:13:38
belfast-ep            SUCCESSFUL    2005.11.14 09:12:54  2005.11.14 09:13:38
```

Once the remove operation completes, you can double check that both the running processes and the Product inventory on the Endpoint, do not show the Universal Agent anymore using the **cinfo** command:

```
[root@belfast][/]-> cinfo -r

*********** Mon Nov 14 13:23:07 CST 2005 ******************
User      : root         Group: system bin sys security cron audit lp
Host name : belfast      Installer Lvl: 400 / 100
CandleHome: /opt/IBM/ITM
***********************************************************
Host     Prod    PID     Owner   Start    ID  ..Status
belfast  ux      20642   root    09:46:50 None ..running

[root@belfast][/]-> cinfo -i

*********** Mon Nov 14 13:23:08 CST 2005 ******************
User      : root         Group: system bin sys security cron audit lp
Host name : belfast      Installer Lvl: 400 / 100
CandleHome: /opt/IBM/ITM
***********************************************************
...Product inventory

ax      IBM Tivoli Monitoring Shared Libraries
         aix513  Version: 06.10.00.00

jr      Tivoli Enterprise-supplied JRE
         aix513  Version: 400 Rel: 100
```

```
ui      Tivoli Enterprise Services User Interface
         aix513  Version: 06.10.00.00

ux      Monitoring Agent for UNIX OS
         aix513  Version: 06.10.00.00
```

You can now remove the OFFLINE entries in the IBM Tivoli Enterprise Portal Workspace related to Universal Agent (um), as detailed in 6.11.6, "Remove the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint from the TEP Workspace" on page 365.

Monitoring Agent for UNIX OS

Now that the Universal Agent has been uninstalled successfully, if you want to completely remove the IBM Tivoli Monitoring 6.1 code from the endpoint, you can use the same procedure.

At the end of the remove process, the CandleHome directory and all the references to the previous installation will be removed.

To clean up the IBM Tivoli Enterprise Portal Workspace 6.1, refer to 6.11.6, "Remove the Monitoring Agent for IBM Tivoli Monitoring 5.x Endpoint from the TEP Workspace" on page 365.

12

Advanced tuning and configuration

In this chapter, we cover some advanced configuration and tuning topics related to IBM Tivoli Monitoring 6.1, including the following topics:

- Building a well-performing IBM Tivoli Monitoring 6.1 environment
- Tuning the running monitoring environment
- Hub TEMS Hot Standby
- Catalog files synchronization

12.1 Building a well-performing IBM Tivoli Monitoring 6.1 environment

The successful IBM Tivoli Monitoring 6.1 implementation is gaining importance for a growing number of departments and users within a company. As the overall IT function is increasingly being run as a business and gets more exposed to other functions within the company, concepts such as IT availability and IT performance become critical to the overall business. IBM Tivoli Monitoring 6.1 in its own right is critical to the successful implementation of these concepts and hence to the functioning of the overall business.

As with all projects, good preparation determines the outcome of the IBM Tivoli Monitoring 6.1 implementation: clear objectives and targets do more than just help you to run the project; they are prerequisites.

Good project preparation also determines whether the resulting application will perform well: if it is well known what information has to be provided in the end and to all its users, the application can be built to the requirements. Although the end result certainly must be open and flexible to answer on demand data needs from the user, building a user interface that is too open or too flexible only results in poor performance and data overload, which will not satisfy the end users: If an end user receives too much data and cannot quickly find the data he requires to do his job—and the data is received late—then the installation has missed its target and can be called a failure. Finding the right mix is the “art and magic” of the project manager.

12.1.1 Preparing the implementation

Today’s monitoring applications are so powerful and can generate such a variety of information in such diverse visualization formats that making the right choices about what is required from a business perspective is not just an option, but a requirement.

The first action is to determine who the users are.

IBM Tivoli Monitoring 6.1 users

The times are long gone that only systems programmers, and optionally operations, were using monitoring data. Many other users now require information, for which the source lies with IBM Tivoli Monitoring 6.1. At this time, it is not important whether end users need the data on IBM Tivoli Monitoring’s own portal TEP or as derived data in applications such as IBM Tivoli Service Level Advisor, Tivoli Data Warehouse, home-grown applications, and others. A full inventory of users (in terms of user departments) should be created.

Determine data needs

For every user department, a senior representative should be chosen as the focal point for the project. This person will represent the department and should be aware of the needs they have regarding systems management information.

From meeting with the representatives, a full inventory of the needs should be built. These needs include the following categories:

Reports	Define the data that should be provided on screen, in paper reports, or to another application for further processing.
Alert	What kind of events are to be reported for every kind of user?
Action	Which automated actions should be taken?

For every piece of data, the inventory should include the time spans the data has to be collected and at what intervals it should be collected. Also, the data to be collected should be specified by attribute, such that all unnecessary attributes can be eliminated from data collection and all unnecessary data transfer across the network is avoided.

It is within the role of the Project Manager or assistant to negotiate the data requirements with the end-user departments. During this phase it is important to challenge the end users to provide as much detail about their requirements as possible and to push for solutions that will decrease the overhead of the overall monitoring application.

Reports

It is better to develop more reports but to make sure that the reports focus on the needs of the end user, rather than to create fewer reports that generate large result sets. The selection of the report can be made by the end user using Navigator items and workspace selections on a single Navigator item.

Situations

Determine the timeframes within which the situation should run. Also try for the longest sampling intervals possible, as this can have significant impact on overhead. Short situation sampling intervals (1 minute or less) should-be reserved for very critical applications and their most critical alerts. The situations should be prioritized by the end user department; this may be used later in the project, for example, to reduce the number of situations if overhead is unworkably high or to add situations or shorten their interval if there are sufficient resources to do so.

Historical data collection

Besides the obvious trimming of the data (only collect the data for which there is a real business need), discussions should focus on how often to collect it. Again, if possible, the longer the collection interval, the less overhead will be needed. Next to the interval, the Project Manager must review how often data should be summarized, when data becomes obsolete, and when it should be removed. A precise inventory should be built. This document should also contain a spreadsheet with a list of all data collected and the planned volumes, based on best estimates and simulations.

Overall, the most common data needs can be divided into:

- Collected at regular interval
  - Alerts generation: For example, operators need to watch critical system and application parameters for exceptions that indicate a threat to availability or poor performance for the end user of an application.
  - Sample Historical Data Collection needs:
    - Capacity planning
    - Service level monitoring
    - Data required for *post mortem analysis* of past problems
    - Trend analysis for workload balancing
    - Accounting and charge back
  - Reports: Reports will be run at regular intervals if an end user sets the auto-refresh option to on at the TEP. This use should be avoided – if a specific system or attribute should be verified regularly, it is better to create a situation for this purpose.
- Collected on demand

  For instance:
  - System programmers: ad hoc detailed reports on systems and applications performance
  - Application development: detailed application tracing to determine applications' resource usage and deadlock prevention

  A good analysis of the data has many benefits, such as:
  - Well-performing monitoring application
  - Low overhead of the monitoring application
  - Users get "focused" data: exactly what they need, so users can do their job effectively
  - No data disruption on Historical Data Collection

12.1.2 Implementing the requirements

After a thorough data analysis has been made, the implementation project team has to set up the entire environment, first on a test system, then install the same environment on production.

After the raw software installation has been done on all systems, a number of out-of-the-box objects are available:

- Physical navigation tree: Shows all nodes and agents that have connected to the TEMS.
- Default workspaces and reports: When selecting nodes, agents, resources, and so forth on the physical navigation tree, the user sees product-provided workspaces that give overview data as well as in-depth monitoring data.
- Alerts: Each agent comes with a set of default situations, some of which will be activated by default, causing alerts to appear on the TEP.
- Automation:Alerts can be linked easily to automated actions.

These objects can be used initially for testing, but are not sufficient for a full implementation. Users will need a more personalized view of the systems, applications, databases, and such that they manage or use.

Because at this point the user requirements are documented, they must be translated into IBM Tivoli Monitoring objects: workspaces, views, situations, and policies. Either of following approaches can be used to further the analysis.

Reports

The first step is to define the logical views: IBM Tivoli Monitoring 6.1 software comes with the physical view, which normally is not used by end users except possibly Systems Programming. Best practice is to build logical views that group the reports that end users require into a logical navigation. In many cases, a single logical view will suffice for a specific group of users. If needed, multiple views can be built if either the logical view is too large for easy viewing or the team of users is functionally divided into several subteams.

After the choice of logical views has been made, these should be subdivided into more detailed views or reports. Each of these subviews will represent one or more workspaces.

Finally, workspaces are again subdivided into views that contain a single report. For every view, the attributes to show must be detailed. Even more important are the filtering criteria: Showing large tables of data is not normally what the end user requires. Instead, if the report represents a large number of rows to display, the single view should be subdivided into several small views. Each view should focus on a small selection of the total and should be directed toward a specific information need of the end user.

Here is an example to clarify this point.

Suppose that we are developing a logical view (or Navigator View, as it is called on the TEP) for WebSphere MQSeries® Administrators. This will be their top-level view.

These MQSeries Administrators may need a lot of views to facilitate their job. One of the reports could be a list of all queues, defined at a Queue Manager. At some Queue Managers, there may be thousands or even tens of thousands of queues. Providing a report that lists these tens of thousands of queues is not very helpful to the Administrators and could generate a lot of overhead to the Monitoring Environment and delays for other users.

As a first step, you should define selection criteria to narrow down the report to a number of queues that will interest the Administrator, such as the top 10 queues with the highest number of messages on the queue. Based on the requests received from the end users, you can now build several focused views. Under the MQSeries Administrators top view, you can now create several workspaces that all provide a specific report to the Administrator.

All further data requirements from MQSeries Administrators can be processed in similar fashion: Under the main MQSeries Administration logical view, the requests should be grouped in more detailed Navigator items (several levels can be created) and per Navigator item, several workspaces can be defined, one being the default workspace. Each workspace can group a few related views.

Building your reports this way ensures that end users get focused views with easy, fast navigation to the data they require while preserving resources and ensuring that the monitoring application performs well.

Here are some more best practices to be used for reporting:

- When creating views, avoid using the thresholds selection in the view. A better practice is to create a query that focuses on the data to be displayed: Build the query such that the necessary selections are made at the query level and only the required attributes are returned. When making the selection at the query level, rather than by setting these on the Thresholds tab, you actually ask the TEMA to return only a limited result set (using less CPU and storage at the TEMA, TEMS, and TEPS and reducing network overhead). Otherwise, if the selection is made on the Thresholds tab, the TEMA is asked to return a larger (possibly much larger) result set, and the actual selection of the data for visualization will only occur at the TEPS level.

- The only exception to the rule above is when you build a workspace that has several views that use very similar data: If a single query can be used for all views on the same workspace, that query will be executed only once. The individual views can then further select a subset of the query's result set and customize them for additional visualization. Using multiple queries on the same workspace slows down the response time for the end user because the queries will be executed one by one. Also overhead would increase.
- Distribution of Navigator items on the logical view should be limited to the systems for which the reports are required. Running the queries and collecting the data for systems that are not of interest for the end user is a waste of resources. To ease distribution, you are recommended to create user Managed Systems Lists that group systems in the way that end users look at them. For example, a user MSL can be created for all Windows database servers or for all servers that run a specific application.
- TEP paging is an important consideration. By default, the TEP displays reports in pages of 100 rows. It is recommended that you keep this setting. If more rows are allowed per page, or if no paging is selected, the TEP will need more storage on the client side to handle all of the data. This will start OS paging if sufficient real storage is not available and will greatly slow down the performance of the TEP. Ideally, reports should not exceed 100 rows.
- Be aware that all pages are stored at the TEPS. If your query returns a large result set, it will trash memory at the TEPS. Because the TEPS has to service multiple end users, it can quickly run out of storage and start OS paging. This situation is detrimental to response times.
- Refrain from using auto refresh on report panels. If you want to regularly check whether a specific attribute is within an acceptable range, do not use reports to check this; instead, create a situation to watch over the resource. The situation will be executed at the lowest level component, at TEMA if possible, and will not require overhead at the TEMS and TEPS at every interval, as a report update would do.
- Specifically, if multiple users monitor the same resources or workload, situations should be created to actually watch the performance. That way the detail data on monitoring this resource will be created only once (by the situation) and not multiple times (for every user).

Overall, the impact of reports is unpredictable and fully depends on actions taken by end users—the data is mainly collected on demand. If reports are not created using these guidelines, end users will wait unnecessarily and system resources will be wasted.

If your requirements include providing large reports (result sets of thousands of rows), you may want to look at alternative methods. The components that suffer most from formatting large reports are TEPS and TEP. By making use of the open interface through Web Services (SOAP), you can offload the visualization workload and use other tools to present the data in any required format.

12.1.3 Historical component

The Historical component has two main parts: data collection and data reporting.

Historical Data Collection

After all requirements have been established during the preparation phase, these have to be implemented using the TEP. By far the most important phase in setting up Historical Data Collection is the preparation: clearly identify the data to be collected and at which interval or granularity. Data for which there is no business need established should not be collected.

Data warehousing is best run every hour to avoid having to handle large full day's data volume if the interval is set to once a day.

Next, the intervals at which data are to be summarized must be agreed on with the end users. IBM Tivoli Monitoring 6.1 supports aggregation by the hour, day, week, month, quarter, and year. Frequent aggregation reduces the need to store the detailed data, which normally takes the bulk of storage space. The summarization itself does not affect the performance of TEMAs or TEMS, but may affect the performance of the relational database system if large volumes are handled. The aggregation can best be scheduled during off hours.

As to the configuration: The Warehouse Proxy (only one can be configured in a single monitoring environment, using a single Hub TEMS) should be in the same network as the Hub TEMS, where most TEMA's can connect without a firewall and over a well-performing network. It is best to install this Warehouse Proxy on a powerful Windows server where the Database Manager runs. If the database server is local to the Warehouse Proxy, it can handle significantly higher volumes of data compared to a remote database server.

Historical Data Reporting

Historical Reporting can be done using either IBM Tivoli Monitoring 6.1 (required if you want to include recent data that have not been warehoused yet) or external tools. When using external tools that extract their data directly from the RDBMS system, the impact on the monitoring environment is minimal: Only when the database server is heavily stressed by reporting requests will the TEMA incur delays to send its data to the Warehouse Proxy, because it may not be able to insert them into the database fast enough. This is the preferred way to run Historical Reports. Warehousing is usually done every hour, so only the data for the last hour would be missing.

When using the TEP to generate Historical Data Reports, the same recommendations apply as to real-time reporting: The better the historical reporting requirements have been qualified, the better the reports will answer the needs of the end user and will cause the lowest overhead to the monitoring environment. Historical Reports using non-warehoused data affect TEMS and TEMA, as the data is normally stored at the TEMA and user's request has to be forwarded from the TEP to TEPS, to (multiple) TEMS and finally to the TEMA. If the report returns a large result set, there will be noticeable overhead at all levels, including network traffic.

When running Historical Reports on the TEP against warehoused data, you still cause considerable overhead to the most vulnerable components, the TEP and TEPS, to the extent that other monitoring users will notice the additional workload through increased response times. Again, the preferred method to report these data would be by using external tools that request the data straight from the database server.

12.2 Tuning the running monitoring environment

Even the best-prepared monitoring environment setup eventually requires adjustments to continue to perform well. Because today's monitoring systems have many variables (number of users, situations, policies, and reports to be collected can vary greatly), the monitoring environment itself should be considered as a standard application when it comes to tuning the implementation. The standard OS monitoring agents can be used to collect real-time and historical performance data regarding the TEMAs, TEMS, and TEPS, just as for any other application. The main resources that should be monitored are CPU, storage, and network performance, and for some components disk/dasd performance as well.

Because IBM Tivoli Monitoring 6.1 components often run on a large number of platforms and systems, the most important task in case of poor performance is to isolate the bottleneck component. If you notice poor performance on your TEP, the first action should be to check whether this is an overall condition, occurring for all kinds of agents, across all systems. If this is the case, the delays are probably caused by one of the common components, such as the Hub TEMS or TEPS. If the delays occur only on a subset of systems, this may be related to a specific remote TEMS. If the delays are only for a specific system, the delay is likely to be found at the TEMA (or TEMAs) on that system, but a more in-depth analysis often will be required, covering multiple components and platforms.

12.2.1 Tuning a TEMA

If the bottleneck is located at the TEMA, the local OS Monitoring Agent can usually provide an indication of the problem cause. For instance, on z/OS, IBM Tivoli Monitoring 6.1 OMEGAMON XE on z/OS can be used to review in detail how the Agent uses resources such as storage, DASD, and CPU; even a bottleneck analysis can be performed to detect whether any other z/OS workload affects the performance of the Agent. Also, the INSPect function can be used to trace in detail which TCB (task control block) modules and such use most of the CPU—these data can be used to check whether a specific monitoring routine within this agent is using high amounts of CPU—and monitoring parameters for the agent can be adapted to reduce overhead. Finally, the TEMAs can be traced selectively by internal component and you can review their activity from the Agent logs. Typical actions resulting from a TEMA performance review could include:

- Adapting the agents monitoring control parameters to reduce the levels of detail data you are collecting.
- Change the TEP Views and query criteria to reduce the number of data rows to be collected or extend the intervals at which data is collected.
- Review situations—for example, to reduce the sampling intervals—but also to find new ways of monitoring critical resources and workloads, such that you still get timely alerts (but with less overhead) by using alternative attributes.
- Removing redundant situations, policies, and so forth.
- Change Historical Data Collection parms. (For example, extend the collection interval.)

12.2.2 Tuning a TEMS

If the bottleneck is located at a TEMS, tuning becomes more complex. Within the TEMS, many internal components can be active, and you must isolate the poorly performing one. If normal OS monitoring does not show any obvious resource shortages or if you want to see more detail on what kind of activities are running within the TEMS, you can run a Warehouse Proxy trace within the TEMS. Set the following trace for the TEMS (either change the trace parms and recycle the TEMS, or use the TEMS Service Console to set this trace dynamically):

```
ERROR (UNIT:KPX STATE)
```

This generates trace entries to the TEMS log as shown in Example 12-1 on page 677.

Example 12-1 Trace entries to the TEMS log

```
(434FC094.0001-1E90:kpxrpcrq.cpp,685,"IRA_NCS_Sample") Rcvd 3 rows sz 760 tbl
*.WTPROCESS req _Z_WTPROCESS0 <3146136,57671864> node <Primary:T993GWY5:NT>
```

This indicates that the Situation _Z_WTPROCESS (a grouped situation) returned three entries that match the Situation criteria. The Situation runs at the Windows OS TEMA on T993GWY5.

A typical report request on the TEP would generate a result in the TEMS trace as shown in Example 12-2.

Example 12-2 Trace entries to the TEMS log

```
(434FBB96.0000-1E90:kpxrpcrq.cpp,685,"IRA_NCS_Sample") Rcvd 283 rows sz 1148
tbl *.NTDEVICE req  <1048870,13631665> node <Primary:T993GWY5:NT>
```

This way you can easily track how much data (283 rows of 1148 bytes from table NTDEVICE) was returned to a request, from which agents the data is returned (again, T993GWY5 in this case).

A typical agent heartbeat would be registered as shown in Example 12-3.

Example 12-3 Agent heartbeat data

```
(434FBBE4.0002-1A34:kpxreqhb.cpp,684,"HeartbeatInserter") InsertNodeStatus
completed with status 0
```

By looking at the timestamps, you can also check the time needed to collect data. With some practice, you will learn how to quickly determine which activities are taking place in the TEMS and isolate any that cause performance degradation.

Besides the proxy trace, you can run many other detailed traces at the TEMS, depending on the bottleneck you find. All of these provide information that can either help resolve the bottleneck or report the issue to Customer Support for assistance.

Other tools that help you tune a TEMS include OMEGAMON OS Agents, with in-depth tracing of critical components such as CPU, Storage, and DASD.

Typical actions that can result from a bottleneck analysis at the TEMS include:

- Configuration parameter changes
- Changes to situations, policies, or reports such as increasing their collection intervals
- Overall configuration changes (for example, move several agents to another existing or new Remote TEMS.

- OS parameter changes (for example, provide more user segments on AIX)
- Hardware upgrades

12.2.3 Tuning a TEPS

Most of the previously listed guidelines for TEMS and TEMA are valid for the TEPS. While TEMA and TEMS mainly collect and handle the monitoring data, the TEP and TEPS have to handle the workload of presenting the data in formats that are easy for the end user to use. This by itself presents a pretty high workload and can stress the systems it runs on. Monitoring the TEP and TEPS systems is critical: Resource shortage can cause high delays for the end user. Real storage often proves to be a bottleneck: for example, a TEPS on Windows will slow down considerably if the system starts to page. The solution may be to increase real storage, but you can also review what kind of activity is running while storage usage is unusually high. There may be queries that return a large result set; though the TEP user may only see 100 rows (default paging size for queries), there may be tens or even hundreds of pages of data. All of these are cached at the TEPS, costing a lot of storage. Even if multiple users request the same report, the TEPS will cache the data for every user separately. So, knowing the kind of activity that takes place at TEPS/TEP helps you resolve performance issues.

The TEP is really a multipurpose console: It can visualize different kind of objects in a user-friendly way: events, business views, reports, and so on. If the workload becomes too high for a single system, you can look at a few alternatives:

- Use more-specialized visualization tools. For example, if the TEP has to handle too many events, you can best migrate to IBM Tivoli Enterprise Console, which is a specialized event console. Many and complex Business Views can be migrated to TBSM, or you can even opt to run large reports not on a TEP, but use your own reporting software, extracting the required data over Web Services (SOAP).
- Implement an additional TEPS.

If you choose to implement multiple TEPSs connected to the same Hub TEMS, you may prefer, instead of duplicating the TEPS, to create “specialized” TEPS: Have Systems Programming use a TEPS different from Operations or Business View users. This can help you to avoid having to manually synchronize TEPS if all user data needed for visualization is stored in an RDBMS, and, for example, if you change a view on the first TEPS, you should extract the data from the database and migrate it to the other TEPS databases. By using specialized TEPS, you may avoid having to manually synchronize them as the views and other objects would be different across TEPS.

12.3 Hub TEMS Hot Standby

If your monitoring environment requires high availability, you should look at a few alternatives for fail-over scenarios. The most common ones include a Cluster Server, TEMS Hot Standby, or *Moveable Hub*. We do not cover Cluster Servers (such as HACMP™, Windows Clusters, Veritas) in detail, because many other publications already cover this terrain. If you are a knowledgeable regular user of Cluster Servers, this may be your best option as it fits your company's practices. Otherwise, if you run the Hub TEMS on distributed platform, you may choose to implement Hub TEMS Hot Standby.

12.3.1 Hot Standby: overall process

Overall, running the Hub TEMS in Hot Standby means that you are running a second Hub TEMS (TEMS2) in parallel to the current Hub (TEMS1). Agents or Remote TEMSs must be configured first to be able to connect to two Hub TEMSs. If no agents connect straight into the Hub TEMS (as recommended for larger configurations) but always connect to a Remote TEMS, then none of the agents should be reconfigured. Only the Remote TEMSs requires a double hub connection setup.

Suppose that the TEMAs or Remote TEMS are already running, before the Hub TEMS are started. At their initialization, TEMAs and TEMS will try to connect to the TEMS that is specified first in their configuration. If the connection fails, at next interval, the TEMA or Remote TEMS will try to connect to the other Hub TEMS, and so on.

The first Hub TEMS (TEMS1) to start automatically becomes the Primary Hub in the configuration. You should not start the Secondary Hub within 10 minutes (or 3 minutes if only Remote TEMSs connect to the hub) after the Primary started; otherwise, Agents or Remote TEMS may first try to connect to the Secondary Hub. (Agents and Remote TEMS will try to connect into any available hub, and if none are available, try again at normal heartbeat interval, which by default is 10 minutes for a TEMA and 3 minutes for a Remote TEMS.) When TEMA or Remote TEMS connect to the Secondary Hub, the hub tries to reroute them to the Primary Hub, and this will cause some additional delays at startup.

When both Primary and Secondary Hubs are active, they connect to each other and the Primary shares updates with the Secondary. New objects and changes are passed on from the Primary to the Secondary Hub. This is the normal way of working during standby: The Primary Hub processes all normal hub activities while the Secondary is busy just keeping up to date with changes.

When the Primary Hub (TEMS1) eventually fails, the Secondary Hub (TEMS2) detects communication failures with TEMS1. TEMS2 then tries to confirm that

TEMS1 is down. Within a few minutes it should establish that TEMS1 is down. At that time, the TEMS2 issues a message confirming its change of status and becomes the new Primary Hub. TEMS2 is now also receiving connections from TEMAs, Remote TEMSs (or both) that make the switch to the new Primary Hub. TEMS2 also restarts situations at this time. All sampled situations are being reevaluated and raised if needed. Pure events may not be raised, if there is no longer a source for the alert.

Agents and Remote TEMSs will detect the “Hub down” condition when making their heartbeat connection to the hub. At that time, they try to connect to the Secondary TEMS that was defined during their configuration. If the Agent or Remote TEMS does not get a response from the Secondary either, it waits an interval and tries again to connect to the first hub. This process continues until a hub can be connected or their configuration settings tell them to stop trying.

When TEMS1 is eventually restarted, it reconnects with TEMS2, which is now still the Primary Hub. TEMS1 remains the new Secondary Hub. TEMS2 will now forward updates to TEMS1 to keep it in sync.

12.3.2 Hot Standby limitations and how to bypass them

When deciding for a Hot Standby implementation, you should be aware of its limitations.

First, Hot Standby will not run on z/OS. If you require this, you should be looking into the Moveable Hub implementation as your alternative.

Next, TEMS Hot Standby is in no way an immediate (in microseconds) or user-transparent solution; a typical failover scenario takes from a few minutes to even 30 minutes to complete, depending on options taken during the implementation. As usual, the faster you want the failover to complete, the more it will cost in terms of overhead.

At failover time, end users see a pop-up message on the TEP signalling that the TEPS has lost contact with the Hub TEMS. This is because the TEPS currently does not yet support TEMS failover. Configuration changes may be taken to alleviate the impact to end users, but at the very least they will be required to log on again.

Finally, events that are open when the Primary Hub TEMS comes down may not come up automatically after the Secondary Hub has taken over. When the Secondary Hub detects that it is the new Primary Hub, it restarts the situations. This causes the sampled situations to be redriven, and raised if the alert is still valid, but pure events will not be raised automatically, so you will have to take steps to ensure they are raised again.

12.3.3 z/OS Moveable HUB

TEMS Hot Standby is not supported on z/OS. As an alternative, there is a solution available from IBM Services. While this is not a true Hot Standby solution, it will automate the start-up of an alternate Hub TEMS on another LPAR when the Hub TEMS on the production LPAR is brought down for maintenance or if the system is not available. You can contact your IBM Services department or your IBM representative for more details.

12.3.4 TEPS support

At this time, the TEPS does not support a TEMS failover scenario yet. This feature is planned for delivery with IBM Tivoli Monitoring 6.2, but cannot be firmly committed yet.

Standard procedure requires you to bring down the TEPS when the Primary Hub TEMS is shut down and the Secondary takes over, then reconfigure the TEPS to connect with the new Primary (TEMS2) and restart. Including the saving of the contents of the Relational TEPS DB, this process requires several manual actions and takes about 10 to 15 minutes to complete.

A few alternative approaches can be used to bypass this, including the use of Cluster Servers for the TEPS. Figure 12-1 on page 682 shows an example implementation using a Workload Balancer.

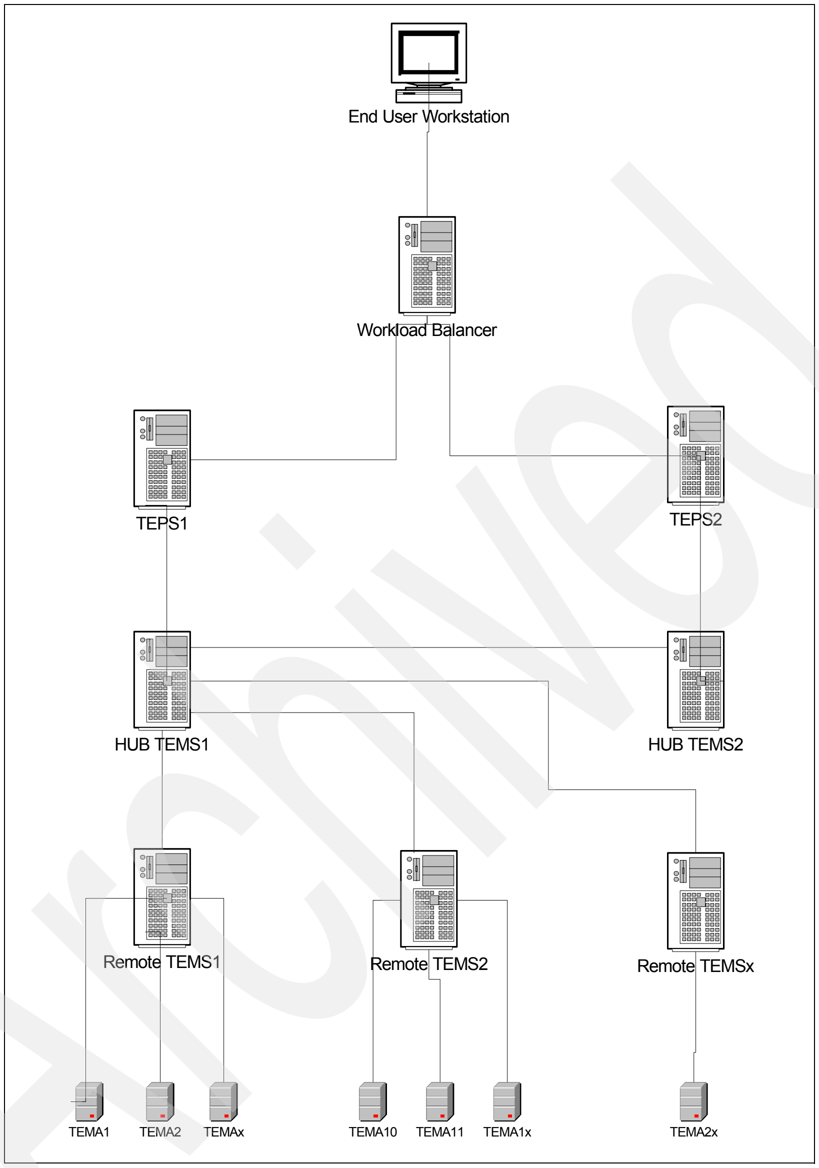


Figure 12-1 HotStandby configuration

12.3.5 Setting up the TEPS for Hot Standby

The new device in this configuration is the Workload Balancer. This can either be a dedicated hardware device or a PC, running a specialized software package. We used WebSphere Edge Server for this setup. The device or software should be configured to forward all network traffic on its ports 1920 and 15001 (or otherwise, as configured in "Configure TEPS Interfaces" on Manage Tivoli Enterprise Monitoring Services) to TEPS1. When TEMS1 then fails, TEMS2 must run a policy that will issue a command when TEMS2 becomes the Primary Hub. The command to be executed should take an action to signal the Workload Balancer to switch from TEPS1 to TEPS2. In our sample setup, we used a situation XYZ, which detects the TEMS2 takeover. Situation XYZ is defined at enterprise level as a TEMS Monitored Application, using the Attribute Group "Managed System." The following attributes are tested:

- Name (INODESTS.NODE): name of a host that should be always online
- Managing System (INODESTS.THRUNODE): name of the Secondary Hub (TEMS2)
- Status (INODESTS.O4ONLINE): *ONLINE

The situation is distributed to the *HUB list. The sampling interval should be set short (for example, 1 minute).

Next, a policy should be created. The first activity in the policy is "Wait until Situation x is true," which should be followed by a Take Action. In our setup, we used **`echo aaaa >> X:\TEMS2.txt`** in which the output file TEMS2.txt is being written to a network drive, on which the Workload Balancer device has access. Depending on the kind of Workload Balancer you are using, it should be capable of catching the result from the Take Action. Best practice is to issue a second Take Action from the policy to remotely start the TEPS2 Services. To function correctly, this policy should be distributed to *HUB and to the host on which the node used in the situation, is running.

The Workload Balancer must be configured to route all requests on its 1920 and 15001 ports to TEPS1 initially, while TEMS1 is the Primary Hub. After receiving the Take Action signal from the policy above, the routing must be reconfigured to reroute all traffic on ports 1920 and 15001 to TEPS2.

A second situation and policy now must be developed to trigger the Workload Balancer to switch back from TEPS2 to TEPS1. The setup is similar to the above.

12.3.6 Sample scenario

In the configuration above, the end users start their TEP session by initiating a browser and pointing to the URL `http://workloadbalancer:1920`. The Workload Balancer will then forward the request to the active TEPS. Now the user can log on to TEPS without even being aware of which TEPS the connection is to.

At TEMS1 failure, TEMS2 takes over. The situation and policy described above will trip and send the message to the Workload Balancer to switch TEPSs and send the command to start TEPS2.

At this next transaction, the end user will be asked to log on to the TEPS.

Keeping the TEPS in sync

Unlike with the TEMS Hot Standby, at current status, the TEPS servers do not keep each other in sync when objects such as workspaces, queries, or views are being updated. This means that when a TEMS failover occurs, the TEPS that takes over may not have the latest levels of these objects.

To ensure that the backup TEPS is in sync:

- Best practice is to disallow all updates on the production TEPS so that all changes are applied first at the test environment. When testing is complete, these objects are migrated to production. This requires regular copies of the RDBMS contents from test to production.
- At regular intervals (daily or whatever is required), the contents from the Primary TEPS are extracted (`migrate-export.bat` will perform this) and moved over to the Secondary TEPS.

> **Important:** The Secondary TEPS should not be running when the new data is inserted.

12.3.7 Pure event support

At failover, the new Primary TEMS will restart all situations. Sampled situations will trip again if the problem condition still exists. True events, however, are not resampled and not redriven. For example, if a monitored log contains an error message that triggered an Alert while TEMS1 was active, this alert will not automatically reappear after TEMS2 has taken over.

If you need the pure events to be raised again after a TEMS takeover, you must develop a bypass. No easy automation is available here. Depending on the type of configuration you run, you may find easier bypasses to raise pure events again.

A possible way out is to log all pure events when they occur: the Situation that detects the event will then need an additional action to write the most important attributes to a file (for example, `ECHO values >> file.txt`). This requires a manual check of all events after a failover occurs.

If you require further automation, you could write a tool, started by a Take Action at failover, that reads the kdsmain.msg file from the previous Hub TEMS, checks for all opened and closed events. and finally writes the still-open pure events to a separate file. This file on its turn should be read by a Universal Agent and should raise new alerts through dedicated situations.

12.3.8 Hot Standby – Tuning parms

The speed at which a Hub TEMS failover scenario will complete can be influenced by changing configuration settings. Speed is affected by:

- Use of the communication protocol between the two hubs. The selection of the protocol to be used for Hub-to-Hub communications can be selected separately during hub configuration. When selecting IP (UDP), you are basically using a connectionless protocol. When the Primary Hub goes down, the Secondary Hub has to send a heartbeat to the Primary Hub, await the failure, then react. This process will likely take several minutes. By specifying IP.PIPE (TCP), the two Hub TEMSs are continuously connected and the Secondary Hub will detect a Primary Hub down within seconds. *IP.PIPE is therefore the protocol of choice.*
- CTIRA_HEARTBEAT: This variable is set at the TEMAs and Remote TEMSs. It defines the interval at which the TEMA or Remote TEMS will send its Heartbeat to the hub; default is 10 minutes for the TEMA (value set to 600). The TEMA detects a Hub-down condition only if the heartbeat is not accepted at the hub. By setting the heartbeat to a shorter interval, the TEMA will switch to the new Hub faster after the Primary fails. This parm should be used with care because it increases overhead on network traffic and at the TEMS that will need to handle more heartbeat calls. This value should not be set lower than 2 or 3 minutes. Also, if you have TEMAs or Remote TEMSs that have a higher priority than others, you can set the heartbeat interval shorter for these, but leave it at 10 minutes for the others. Note that changing this parm will not influence the interval at which the TEMS reports a TEMA as “offline”: the interval changes the time at which the TEMA sends the heartbeat, but the TEMS will still check only every 10 minutes if any TEMA or Remote TEMS has not updated its Heartbeat row and should be reported offline.
- CTIRA_MAX_RECONNECT_TRIES: Maximum number of times that the TEMA tries to connect to a TEMS. After this number of tries, the TEMA stops. Default is 720 times.
- CTIRA_RECONNECT_WAIT: Interval the TEMA waits before trying to reconnect to a TEMS. Default is 600 (10 minutes). Using the defaults for CTIRA_MAX_RECONNECT_TRIES and CTIRA_RECONNECT_WAIT, the TEAM will stop trying to reconnect after five days.

12.4 Catalog files synchronization

The IBM Tivoli Monitoring 6.1 configuration makes use of Catalog files to define the data that flow across its components. A full description of how these Data Definition Files are composed and used is beyond the scope of this document. For now, we are concerned with an issue that frequently arises in IBM Tivoli Monitoring configurations: de-synchronization of the Data Definition files.

IBM Tivoli Monitoring 6.1 Data Definitions are stored in three types of files (sample directories are for Windows and can differ by platform) and are created by Product Code (TEMA):

- ODI files: These contain the full data definitions and are used to generate the CAT and ATR files. These files are named DOCKxx, where xx represents the Product Code. They can be found in the \HOME\CNPS directory. Universal Agent created ODI files will be called xxxODI00 (where xxx represents the UA Application name, and 00 is for the application version).
- CAT (or Catalog) files: These can be found in the \HOME\...\RKDSCATL directories.
- ATR (or Attribute) files: These can be found in the \HOME\...\ATTRLIB directories.

Both TEMA and TEMS make use of the CAT and ATR files, but TEPS uses only the ODI files. The Data Definitions should always be synchronized between TEMS and TEPS. If an upgrade of the Application code is made to the TEMS, it has to be applied to TEPS and vice versa. TEMA files can be of an older date than at the TEMS and TEPS level. This allows for a staged implementation of new versions and levels: First the TEPS and TEMS should be upgraded simultaneously, and the TEMAs can be upgraded later because this requires actions on possibly thousands of systems.

All too frequently, these files get out of sync. The most common symptom is empty reports on the TEP — or situations with status `in error` — specifically, when some of the Reports for a TEMA are fine, while other reports for the same TEMA fail, you will find these files to be out of sync.

When these files are out of sync, you will also find error messages in TEMS or TEPS log files (or both), indicating Catalog failures.

The easiest way to determine whether files have gotten out of sync is to check the dates and sizes of the CAT, ATR, and ODI files across TEPS and TEMS. Also, the first line of the CAT and ATR files contain a timestamp.

12.4.1 Corrective action

The safest way to again synchronize Catalog files across TEPS and TEMS is to install the same product (application) release and level at all TEPS and all TEMS that make up the same configuration.

If an emergency repair is needed, CAT, ATR, and ODI files can be copied across manually. Care must be taken that these files are not transmitted in binary format, but translated to the target platform.

Special case: Universal Agent

Although this issue is usually created by installing a new product level at the TEPS but not at the TEMS level, or vice versa, it arises more frequently when the configuration comprises one or more Universal Agents.

CAT, ATR, and ODI files for Universal Agents are not part of a product installation, but are generated based on the application definitions by the user. When the user creates or modifies Universal Agent Metafiles, new CAT, ATR, and ODI files are generated at the Universal Agent level. Then when the Agent connects with the TEMS, it pushes the new data definition files to the TEMS and TEPS level. However, the Universal Agent only takes this action once. At this time, there can be intermittent problems and CAT, ATR, and ODI files may not be created or replaced correctly at all levels.

One of the reasons that these files are not distributed correctly across to TEMS and TEPS is if your configuration contains a Hot Standby TEMS. Because the Agent will push the new definitions only once, the Secondary (or inactive) TEMS may not have received the new copies.

Pushing new versions of data definition files may also fail if older versions of these files still exist. If a new metafile is created containing a new Universal Agent Application, then the distribution across TEMS and TEPS usually works well (also if a new version of the application is created). However, if a current version of an existing application is being replaced, problems may arise. Care should be taken to execute the Universal Agent cleanup (um_cleanup) at all levels (TEMA, TEMS, and TEPS) such that all “old” data definition files have been removed.

When CAT, ATR, and ODI files have gotten out of sync, the best action is to copy these over manually across all platforms. Remember that these are text files, not binary.

Firewall configurations

In most IBM Tivoli Monitoring 6.1 implementations, firewalls play an important role. For a successful implementation, it is important to understand the way the components communicate with each other. The entire configuration can be split into two major parts that have to be set up to support firewalls. Most of the components such as TEMA, TEMS, and TEPS use the same protocols, and we describe their functioning later. For the TEP Client to TEP Server communications, extensive firewall configuration data and samples can be found in *IBM Tivoli Monitoring Installation and Setup Guide*, GC32-9407.

12.5 Miscellaneous hints and tips

This section provides technical insight into IBM Tivoli Monitoring that might not be found reading the official product manuals. The tips are categorized by infrastructure component.

12.5.1 Tivoli Enterprise Monitoring Server

The following hints and tips concern the Tivoli Enterprise Monitoring Server:

- Recommendation is to always begin initial installation with implementing the Hub TEMS.
- Install at least one product support component during the TEMS installation.
- Always seed the Tivoli Enterprise Monitoring Server immediately after installing. Remember to reboot the TEMS right after seeding. This prevents blank workspaces in the TEP because the TEMS does not support the attribute groups.
- The Tivoli Enterprise Monitoring Server must be started in order to seed for application support.
- It is recommended not to enable security validation when installing the Hub TEMS initially. Set up the environment completely before enabling security validation within the installation. Remember, `sysadmin` is the default administrator user ID.
- Set the CTIRA_HEARTBEAT variable in the KBBENV file to specify the heartbeat interval among the Hub TEMS and the Remote TEMS. The default interval is 3 minutes.
- Using the AIX platform: the TEMS runs as a standard AIX application that by default makes use of only a single memory segment (256 MB). To solve performance issues, reconfigure the TEMS to use multiple user segments instead of adding additional memory.
- Two types of settings can be used to restrict source (local ephemeral) port usage of socket-based protocols to a specific set of ports or port ranges. Use the following types inside the variable KDC_FAMILIES behind each protocol

definition. The KDC_FAMILIES variable is found in the component environment files (for example, kumenv).

- POOL:portnum
- POOL:lowrange-highrange (range size is limited to 1024 ports)

For example:

```
KDC_FAMILIES=IP.PIPE POOL:5000-5100 PORT:1918 IP use:n SNA use:n
IP.SPIPE use:n
```

How the Hot Stand-by works:

- When there is a failure on the acting Primary Hub TEMS
- When the switch is initiated by a user on a TEMS.

There is no automatic switch done when the primary comes back up. The hot stand-by expects the primary Hub TEMS and the alternate (stand-by) Hub TEMS to be at the same capacity. Internally, hot stand-by considers them peers and does not distinguish in terms of a primary and a secondary. Rather it handles them as acting-primary and acting-secondary. Both Hub TEMSs will alternate between being acting-primary and acting-secondary.

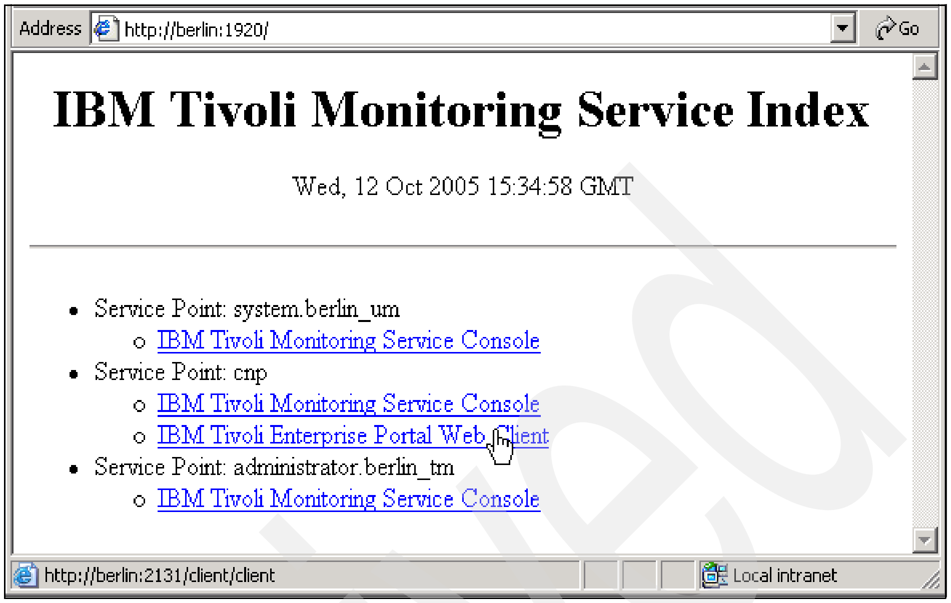
By default, the algorithm that the hot stand-by follows to determine the acting-primary Hub TEMS queries the two TEMSs to determine how long they have been up. The Hub TEMS that has been up the longest wins.

► IBM Tivoli Monitoring Service Console

The service console can be used to remotely access IBM Tivoli Monitoring components for diagnostic information. It is accessed over the generic port 1920 of the CT/HTTP daemon, which chains all the different services running on a single system in the service index.

- Accessing an individual service console from the service index list opens an arbitrary ephemeral port > 1024 for that particular console.
- It requires a valid operating system user ID and password for access.
- Access can be blocked by disabling the internal Web server by adding the parameter `HTTPD use:n` to the end of the KDC_FAMILY variable. But, then a third-party Web server is necessary for the TEPS to function, and all desktop clients have to be reconfigured.

Tip: Accessing the service management console on a TEPS provides a URL to access the Tivoli Enterprise Portal browser client.


Address http://berlin:1920/

IBM Tivoli Monitoring Service Index

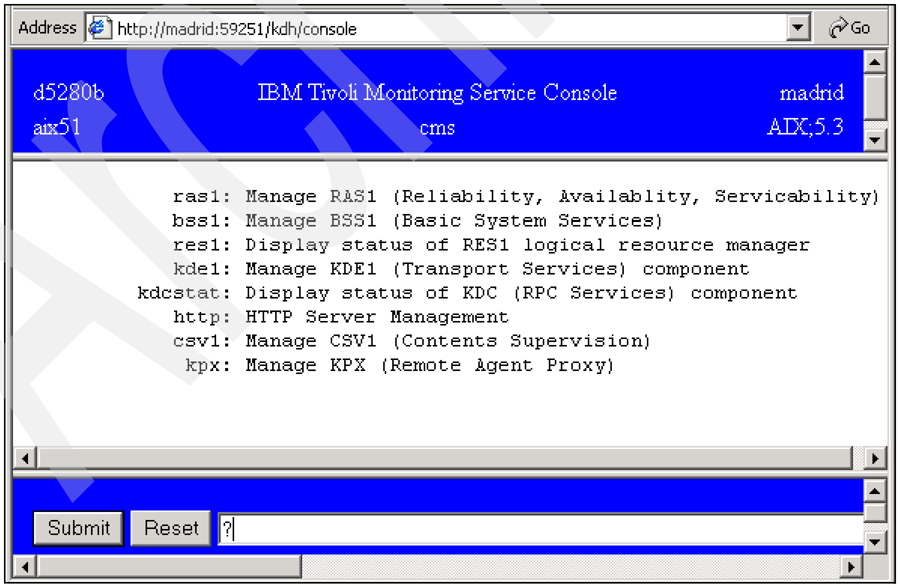
Wed, 12 Oct 2005 15:34:58 GMT

- Service Point: system.berlin_um
  - IBM Tivoli Monitoring Service Console
- Service Point: cnp
  - IBM Tivoli Monitoring Service Console
  - IBM Tivoli Enterprise Portal Web Client
- Service Point: administrator.berlin_tm
  - IBM Tivoli Monitoring Service Console

http://berlin:2131/client/client Local intranet

Figure 12-2 Service management console on berlin - http://berlin:1920

For additional command syntax help, type ? on the service management console input field.


Address http://madrid:59251/kdh/console

d5280b IBM Tivoli Monitoring Service Console madrid
aix51 cms AIX;5.3

```
        ras1: Manage RAS1 (Reliability, Availablity, Servicability)
        bss1: Manage BSS1 (Basic System Services)
        res1: Display status of RES1 logical resource manager
        kde1: Manage KDE1 (Transport Services) component
     kdcstat: Display status of KDC (RPC Services) component
        http: HTTP Server Management
        csv1: Manage CSV1 (Contents Supervision)
         kpx: Manage KPX (Remote Agent Proxy)
```

Figure 12-3 Typing ? in the console input field produces a help page

Tivoli Enterprise Portal Server

The following hints and tips concern the Tivoli Enterprise Portal Server:

- After installing the Tivoli Enterprise Portal Server, immediately install the application support for the portal. The application support is necessary during the portal server configuration to properly create the tables in the database. This prevents blank workspaces in the TEP client.
- Always install the RDBMS locally on the Tivoli Enterprise Portal Server before installing the TEPS product. It is highly recommended to keep this database local to the system.
- If Security Validation is enabled, remember to manage the user IDs at the Hub Tivoli Enterprise Monitoring Server as well as within the Tivoli Enterprise Portal Server.

Tivoli Enterprise Management Agent

The following hints and tips concern the Tivoli Enterprise Management Agent:

- Set the CTIRA_HEARTBEAT variable in the KNTENV file on Windows to control the heartbeat interval.
- Set the CTIRA_HEARTBEAT variable in the agent .ini file (for example, lz.ini, ux.ini, ua.ini). When the agent is stopped and restarted, the agent configuration file is recreated using settings in the .ini file.
- The UNIX Log Agent supports the Windows platform log files as well.
- If the UNIX Log Agent does not find the required log file, it shuts down after 30 seconds.
- The variable CITRA_HOSTNAME controls the presentation of the agent in the navigator tree.

Tivoli Warehouse Proxy agent

Due to the Windows platform limitation, only a total of 1,500 agents can connect to the Warehouse Proxy agent using the protocol IP.PIPE. This can impede the amount of TEMA warehousing historical data within firewall environments.

SOAP interface

IBM Tivoli Monitoring Web Services solution provides an industry-standard open interface into IBM Tivoli Monitoring solutions. This open interface offers easy access to performance and availability data, enabling you to use this information for advanced automation and integration capabilities.

IBM Tivoli Monitoring Web services implements a client/server architecture. The client sends SOAP requests to the IBM Tivoli Monitoring SOAP server, which receives and processes the SOAP requests from the client. Predefined SOAP methods let you perform many functions within the platform environment. You can begin to use SOAP methods immediately, and use them as templates in creating your own advanced methods.

SOAP works with any programming or scripting language, any object model, and any Internet wire protocol. The SOAP methods can be invoked via PERL, JavaScript, VBSCRIPT, JSCRIPT, C++, and through a browser.

Refer to *IBM Tivoli Monitoring Installation and Setup Guide*, GC32-9407, for additional details about installing and configuring the SOAP product.

What is SOAP?

Simple Object Access Protocol (SOAP) is a communications XML-based protocol that lets applications exchange information via the Internet. SOAP is platform independent and language independent. SOAP uses XML to specify a request and reply structure. It uses HTTP as the transport mechanism to drive the request and to receive a reply.

Important: Prior to using the IBM Tivoli Monitoring solution, you must have a basic understanding of SOAP, Extensible Markup Language (XML), XML Namespaces, and Web Services Description Language (WSDL)

IBM Tivoli Monitoring provides numerous SOAP methods for use with the Web Services integration. These methods enable you to query and control installations dynamically.

The SOAP methods can perform the following actions:

- Stop or start policies and situations.
- Retrieve attribute data that you can display in charts or reports.
- Open and close events.
- Make real-time requests for data.
- Issue SOAP requests as system commands in TEPS.
- Generate daily operation summaries.
- Retrieve data in the Tivoli Date Warehouse.

Attention: User access controls for SOAP are through the user IDs created within the Tivoli Enterprise Portal Server. Define the access privileges for querying and modifying data via the SOAP interface.

Two methods for using the SOAP interface:

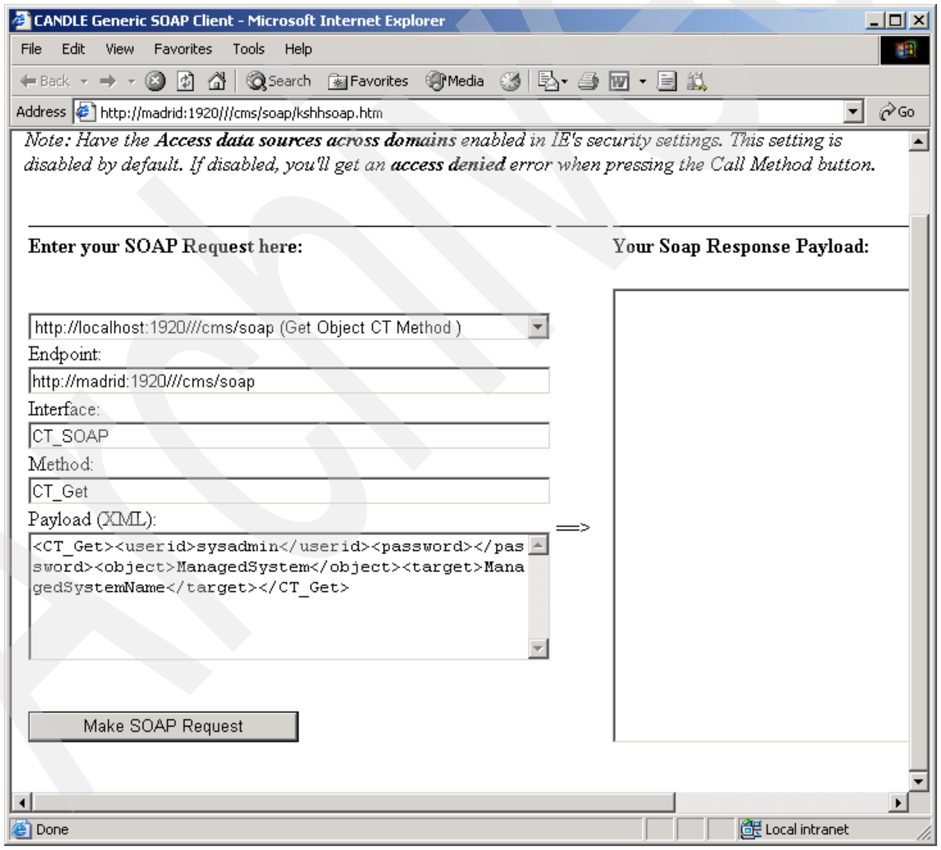
- Internet Explorer
- Using the SOAP client command line utility.

When you use the SOAP client in conjunction with Internet Explorer to issue SOAP requests, you can modify the tags or the text if needed. In contrast, the command line utility simply displays the output of the request at the command prompt.

Figure 12-4 shows a sample Web page using the SOAP interface. This example shows the CT_GET method for IBM Tivoli Monitoring:

```
<CT_Get><userid>sysadmin</userid><password></password><object>ManagedSystem
</object><target>ManagedSystemName</target></CT_Get>
```

Note: Only the method syntax is displayed. A SOAP envelope is necessary to package the IBM Tivoli Monitoring methods in order for the Web services interface to process the request properly.


CANDLE Generic SOAP Client - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://madrid:1920///cms/soap/kshhsoap.htm
Go

*Note: Have the **Access data sources across domains** enabled in IE's security settings. This setting is disabled by default. If disabled, you'll get an **access denied** error when pressing the Call Method button.*

Enter your SOAP Request here:

http://localhost:1920///cms/soap (Get Object CT Method)

Endpoint: http://madrid:1920///cms/soap

Interface: CT_SOAP

Method: CT_Get

Payload (XML):

```
<CT_Get><userid>sysadmin</userid><password></pas
sword><object>ManagedSystem</object><target>Mana
gedSystemName</target></CT_Get>
```

Figure 12-4 An example of the SOAP Web interface using the CT_GET method

Miscellaneous

The following tables list IBM Tivoli Monitoring product codes and IBM Tivoli Monitoring UNIX operating system architecture abbreviations.

Table 12-1 IBM Tivoli Monitoring complete product code listing

Component	Product code	Seed
Active Directory monitoring agent	3z	yes
Alert Adapter for Tivoli Enterprise Console	tv	
Alert Emitter for Tivoli Enterprise Console	vt	
DB2 monitoring agent	ud	yes
TEC GUI Integration	it	
Endpoint monitoring agent	tm	yes
i5/OS monitoring agent	a4	yes
Linux OS monitoring agent	lz	yes
Microsoft Exchange Server monitoring agent	ex	yes
Microsoft SQL Server monitoring agent	oq	yes
Oracle monitoring agent	or	yes
Sybase Server monitoring agent	oy	yes
Tivoli Enterprise Monitoring Server	ms	
Tivoli Enterprise Portal browser client	cw	
Tivoli Enterprise Portal desktop client	cj	
Tivoli Enterprise Portal Server	cq	
Universal Agent	um	yes
UNIX Log Alert monitoring agent	ul	yes
UNIX OS monitoring agent	ux	yes
Warehouse Proxy	hd	
Warehouse Summarization and Pruning agent	sy	
Windows OS monitoring agent	nt	yes

Table 12-2 IBM Tivoli Monitoring UNIX architecture abbreviations

UNIX OS architecture	Abbreviation
AIX v5.1 (32-bit)	aix513
AIX v5.1 (64-bit)	aix516
AIX v5.2 (32-bit)	aix523
AIX v5.2 (64-bit)	aix526
AIX v5.3 (32-bit)	aix533
AIX v5.3 (64-bit)	aix536
HP-UX 11i (32-bit)	hp11
HP-UX 11i (64-bit)	hp116
Linux Intel R2.4 (32 bit)	li6243
Linux Intel R2.4 (64 bit)	li6246
Linux Intel R2.4 GCC 2.9.5 (32 bit)	li6242
Linux Intel R2.4 GCC 2.9.5 (64 bit)	li6245
Linux Intel R2.6 (32 bit)	li6263
Linux Intel R2.6 (64 bit)	li6266
Linux Intel R2.6 GCC 2.9.5 (32 bit)	li6262
Linux Intel R2.6 GCC 2.9.5 (64 bit)	li6265
Solaris v8 (32 bit)	sol283
Solaris v8 (64-bit)	sol286
Solaris v9 (32 bit)	sol293
Solaris v9 (64 bit)	sol296

Tips:

- All UNIX platforms: The file *ciunix.dsc* contains all product descriptions and matching architecture abbreviations.
- The Windows platform product is compressed in .cab files and broken out by product components. It uses unconventional product codes.

13

Administration and maintenance

This chapter covers some basics of administration and suggested maintenance of your IBM Tivoli Monitoring 6.1 environment:

- Administration topics
- Maintenance topics

13.1 Administration topics

Our goal with the administration topics provided in this chapter is to prepare you adequately for the kind of knowledge that you will require to be able to administer your IBM Tivoli Monitoring 6.1 environment. Be sure to read through Chapter 8, “Real-life scenarios” on page 469 to understand how to use the product to do most common types of tasks.

Important: As a part of writing this chapter, the author attended the following Tivoli Education courses:

- IBM Tivoli Monitoring 6.1 for Operators

  http://www.ibm.com/software/tivoli/education/Q956702H03058N69.html

- IBM Tivoli Monitoring 6.1 for Administrators

  http://www.ibm.com/software/tivoli/education/G260590S46948M98.html

The author strongly recommends them to help speed time to value with this product. The topics in those courses cover most of what customers will do in their first implementations of IBM Tivoli Monitoring 6.1.

13.1.1 Idiosyncrasies

To ease the administration of IBM Tivoli Monitoring 6.1, the following paragraphs point out a few idiosyncratic behaviors, some of which are due to the native behavior of Java applications. Note that this Redbook was based on pre-GA code and potentially many of these behaviors could be rectified by GA code, patches, or IBM Tivoli Monitoring 6.2.

Cursor behavior

When using the GUI, the cursor will occasionally fail to change from the “working” hourglass back to your navigational pointer (arrow). If you feel that an action is taking longer than anticipated, be sure to try clicking on the top-most title bar to switch the cursor back to its correct state.

While creating workspaces, after clicking on the toolbar (to give a function to a pane) the cursor will change to a hand. When you move over certain types of pre-existing panes that you wish to replace, the cursor may become an arrow again. This does not indicate that your action was canceled; your cursor still has the concept of the hand and you can continue your desired action by clicking in your destination pane to place the function.

Groups

IBM Tivoli Monitoring 6.1 does not use groups for user administration or workspace management. You cannot establish groups and create members of those groups. The product does not allow any changes to user IDs to be made based on any type of group membership. To assist with user administration, templates are used that facilitate the creation of users.

It is not recommended that shared IDs be used (such as in an operators' center) because the workspaces can be customized so extensively, it would be difficult for several users to coordinate any changes to the workspace.

13.1.2 Product design points

This section describes some design points of IBM Tivoli Monitoring 6.1.

Understanding product features

IBM Tivoli Monitoring 6.1 contains several built-in terminal or remote connection methods, which can be accessed directly from a workspace pane. At this time however, SSH is not one of the support protocols. If this is the desired method to connect to remote machines, a third-party SSH client would have to be used outside of Tivoli.

Due to the number of features in the GUI, many of the buttons provide flyover labels to assist in the identification of each function. Most of the functions also have hotkeys (Ctrl+<letter>) to facilitate an end user's speed to perform tasks.

Because objects in IBM Tivoli Monitoring 6.1 may have several dependencies or things dependent on them, the ability to rename an object is not permitted. In order to rename an object, you must use the "create another" feature, which makes a copy of the original with a new name and allows the deletion of the old object. This ensures any dependencies to that object are updated properly.

Attribute labels all vary in the character limits permitted to the data-type that is scoped in the IBM Tivoli Monitoring code. Most of these limitations are documented (and in some cases, will be increased in future IBM Tivoli Monitoring releases). If you are having difficulty in getting something to work, verify that your label adheres to the character limit. The error message may not always indicate that your label is incorrect.

When configuring the summarization and pruning agent, set the time for the next execution to be at least 10 minutes into the future as it will take a couple of minutes to start. The agent will round the time to the nearest five minutes internally, so setting it forward 10 minutes ensures that the agent starts up and rounds off its next cycle time.

Understanding the security

These terms should be helpful in understanding IBM Tivoli Monitoring 6.1 security.

TEPS

On the browser client URLEnabled mode allows each Tivoli Enterprise Portal to be referenced directly as a URL. The browser mode client includes a URLEnabled parameter, which is set to True by default. This can be changed to disable the workspace URL so that the URL does not change when you switch workspaces and you can use Internet Explorer's Favorites history features.

AGENTS

Agents run as the user that installed the agent.

Linux/UNIX

The agent does not have to run as root (Linux/UNIX). Best practice is to run under a non-privileged application ID. Remember to set up the correct file permissions (**`setperm`**) if you change the user the agent runs as from the one it was installed under.

Windows

The agent does not have to run as Administrator (Windows). Best practice is to run as a non-privileged application account rather than "Local System Account" or "Administrator (SID 500)" on Windows. The agent can be run as another user by setting up the Log On user of the agent service in service control settings.

Certain agents might need special user rights, as described in the corresponding user guides, to access the applications.

Common install problems

Some commonly observed installation problems for UNIX and OS/400 include:

UNIX

```
Do you have root rights for installation? Do not install with root user - use a
user id with all necessary install rights.

Is the DISPLAY variable exported for GUI installation, and does X-Windows (try
running 'xclock') work?

Check if Korn Shell (ksh) is active.

Check available disk space.

Is it a multi homed system (multiple NICs)?
```

OS / 400

```
Do you have QSECOFR rights for install?

Is it a foreign language system?

Are desired ports available?

Do we have to pass firewalls?

Check how TCP / IP name resolution works on that box:
DNS Priority *local or *remote? Should be *local…
SNA System Name in DNS or host table?If not, make sure it is there
```

13.1.3 User administration

This section addresses several topics related to user administration.

Creation of users

The creation of users is done through the administer users panel (Ctrl+u). There is a default user. Changes to this default user will change the defaults when the default user is used to template other users. However, the default user is just a template, and subsequent changes to the template do not affect existing users created from the template.

The default template can be copied by using the option to create another user while the default user is selected. The other option is create a new user, and no settings will be applied to the new user created in this way.

When creating a new or another user, you are prompted to supply three pieces of information for the user. The first, userid, is the only mandatory field you must supply. This user ID is limited to 10 characters with no spaces.

The next field is user name. This is not the same thing as the userid. This field can have spaces and is meant to categorize users. An example might be “ACME operators.” The user name is visible in the users list and can be helpful to organize your users.

The final field is a user description. This field can contain spaces and punctuation. This could be used to store information about the user such as corporate ID number, e-mail, phone number, and so forth.

Previously we stated that there is no concept of groups. There is, however, a concept of templates (such as the default user) that can aid in the creation of user accounts. When you create a “template” account, you should name it in such a way that it is visible to you as a template (Example 13-1 on page 702).

Example 13-1 Template accounts

```
userid: template1 (or tmpltoprns --> remember! limited to 10 characters)
user name: template for creation of operator accounts
user description: this template should be used to create all operators.
```

Just a couple of important points:

- You do not want the template user IDs to correlate to actual users because this would be a security violation. Make sure your system administrators do not create a user account by these names to avoid anyone gaining access to the environment and using template administrators to do things.
- Remember to create another after selecting your template, not create a new user.
- Create all users based on copies of the templates (and thus not default user) or new users. This enables you to stick to your policies for security and job separation that you will design.

Authentication is strongly recommended to meet most corporate security requirements. The authorization of users is activated by using the MTEMS on Windows or **`itmcmd manage`** on Linux/UNIX.

When using authentication for users, this account will be associated with a user name on the TEPS server. This account may be a Windows domain account or a system account. However, the following restrictions apply:

- The account must exist on the TEPS and TEMS servers.
- The account may be a domain account on Windows if the TEPS and TEMS are both Windows.
- The account must use 10 or fewer characters for the user name.
- LDAP is not supported.

The authentication is done against the Windows accounts or domain accounts, RACF® or ACF/2 on z/OS, or /etc/passwd on Linux/UNIX.

The three main areas of definition for a user are permissions, applications, and workspaces.

The Permissions tab enables you to assign some granularity to the capabilities of your users. The philosophy when giving permissions to users is to keep them to minimal permissions for their job.

The Applications tab enables you to control which agents and applications the user has access to.

The Workspaces tab defines the views the user can access. Granularity here only comes from the creation of custom navigator views and removal of access to the physical and logical views. You can also set the default view for the navigator view for this user.

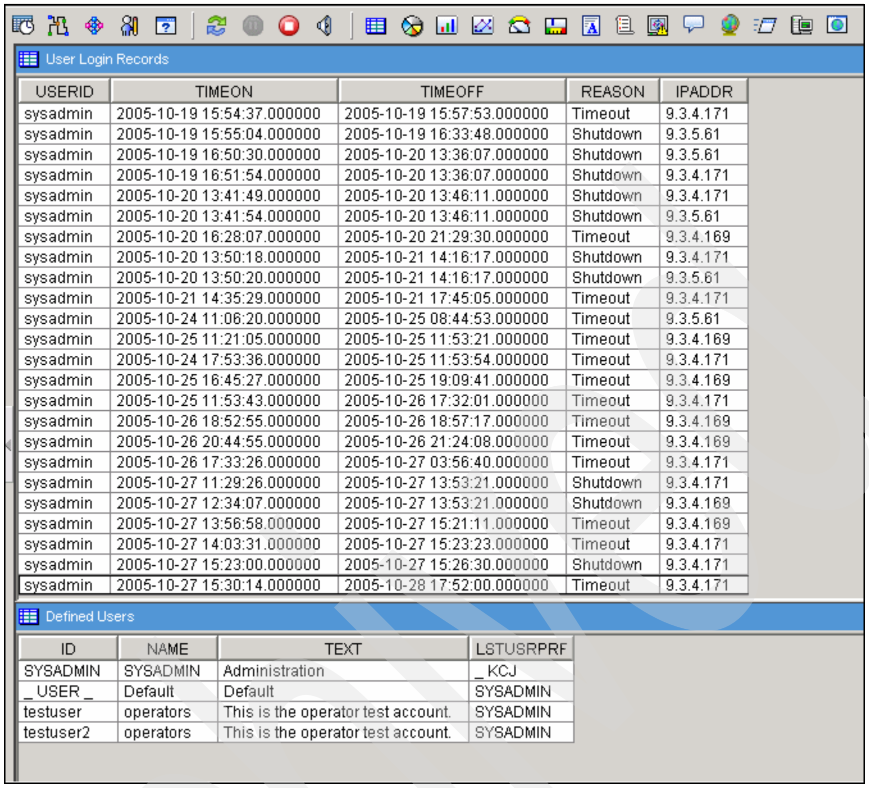
You cannot restrict a user to a set of machines (for example, when using the physical navigator view for this user); however, the default view restricts the user to the view you select here or below.

As an example, you can select UNIX Systems in the physical navigator and restrict the administrator to only UNIX systems (any and all systems in that view are then available), but you cannot choose to restrict them to a single system under UNIX or to give them multiple choices like UNIX and Windows.

To provide a more granular set of limited views for systems or applications would require multiple accounts or a custom workspace with a limited number of systems (or both) and using that instead of the physical or logical navigators for those users.

User maintenance strategy

According to your security rules, you should conduct semiannual or annual security audits of all user accounts. All the information about users and their logons is stored in the TEPS database. An auditor workspace could be created that shows the results of a couple of simple queries as shown in Figure 13-1 on page 704.


User Login Records

USERID	TIMEON	TIMEOFF	REASON	IPADDR
sysadmin	2005-10-19 15:54:37.000000	2005-10-19 15:57:53.000000	Timeout	9.3.4.171
sysadmin	2005-10-19 15:55:04.000000	2005-10-19 16:33:48.000000	Shutdown	9.3.5.61
sysadmin	2005-10-19 16:50:30.000000	2005-10-20 13:36:07.000000	Shutdown	9.3.5.61
sysadmin	2005-10-19 16:51:54.000000	2005-10-20 13:36:07.000000	Shutdown	9.3.4.171
sysadmin	2005-10-20 13:41:49.000000	2005-10-20 13:46:11.000000	Shutdown	9.3.4.171
sysadmin	2005-10-20 13:41:54.000000	2005-10-20 13:46:11.000000	Shutdown	9.3.5.61
sysadmin	2005-10-20 16:28:07.000000	2005-10-20 21:29:30.000000	Timeout	9.3.4.169
sysadmin	2005-10-20 13:50:18.000000	2005-10-21 14:16:17.000000	Shutdown	9.3.4.171
sysadmin	2005-10-20 13:50:20.000000	2005-10-21 14:16:17.000000	Shutdown	9.3.5.61
sysadmin	2005-10-21 14:35:29.000000	2005-10-21 17:45:05.000000	Timeout	9.3.4.171
sysadmin	2005-10-24 11:06:20.000000	2005-10-25 08:44:53.000000	Timeout	9.3.5.61
sysadmin	2005-10-25 11:21:05.000000	2005-10-25 11:53:21.000000	Timeout	9.3.4.169
sysadmin	2005-10-24 17:53:36.000000	2005-10-25 11:53:54.000000	Timeout	9.3.4.171
sysadmin	2005-10-25 16:45:27.000000	2005-10-25 19:09:41.000000	Timeout	9.3.4.169
sysadmin	2005-10-25 11:53:43.000000	2005-10-26 17:32:01.000000	Timeout	9.3.4.171
sysadmin	2005-10-26 18:52:55.000000	2005-10-26 18:57:17.000000	Timeout	9.3.4.169
sysadmin	2005-10-26 20:44:55.000000	2005-10-26 21:24:08.000000	Timeout	9.3.4.169
sysadmin	2005-10-26 17:33:26.000000	2005-10-27 03:56:40.000000	Timeout	9.3.4.171
sysadmin	2005-10-27 11:29:26.000000	2005-10-27 13:53:21.000000	Shutdown	9.3.4.171
sysadmin	2005-10-27 12:34:07.000000	2005-10-27 13:53:21.000000	Shutdown	9.3.4.169
sysadmin	2005-10-27 13:56:58.000000	2005-10-27 15:21:11.000000	Timeout	9.3.4.169
sysadmin	2005-10-27 14:03:31.000000	2005-10-27 15:23:23.000000	Timeout	9.3.4.171
sysadmin	2005-10-27 15:23:00.000000	2005-10-27 15:26:30.000000	Shutdown	9.3.4.171
sysadmin	2005-10-27 15:30:14.000000	2005-10-28 17:52:00.000000	Timeout	9.3.4.171

Defined Users

ID	NAME	TEXT	LSTUSRPRF
SYSADMIN	SYSADMIN	Administration	_KCJ
USER	Default	Default	SYSADMIN
testuser	operators	This is the operator test account.	SYSADMIN
testuser2	operators	This is the operator test account.	SYSADMIN

Figure 13-1 A sample auditor workspace

The information used by these reports is contained in two TEPS tables: KFWUSER and KFWJRNLLOGIN. The information retrieved is obtained via the TEPS server, so it might be better obtained via some database reporting tool such as Crystal Reports. There is no easy way to extract permissions information about the user's privileges from this table, so auditing must be done via visual scan of the Administer Users panel.

Table 13-1 on page 705 suggests what types of users should have access to which privileges based on the following role definitions:

- ITM SuperUser is an administrator who customizes and manages the infrastructure framework.
- ITM Admin is an administrator who manages user account creation.
- Console Operator is someone who views the event console and perhaps has permission to customize their workspace.
- Report User is someone who only views the report information.

- Application Admin is an application administrator for a managed application.
- System Admin is a system administrator who manages the managed systems.

Each of these permissions is described in *IBM Tivoli Monitoring Administrator's Guide*, SC32-9408. Also, certain permissions are sometimes required for other permissions; for example, removing view also removes modify. The product enforces this when you use the Administer Users panel.

Table 13-1 What types of users should have access to which privileges

	ITM SuperUser	**ITM Admin**	**Console Operator**	**Report User**	**Application Admin**	**System Admin**
Action: view	x		x		x	x
Action: modify	x					
Agent: manage	x		x		x	x
Agent: stop/start	x		x		x	x
Customize Navigator Views: modify	x		x		x	x
Event: close	x		x		x	x
Event: view	x		x	x	x	x
Event: acknowledge	x		x		x	x
Feature: DE: enable						
History: configure	x					
Launch App: launch	x		x		x	x
Launch App: view	x		x		x	x
Launch App: modify	x		x		x	x
Managed System List (MSL): view	x		x		x	x
MSL: modify	x					x
Policy: view	x		x		x	x
Policy: stop/start	x		x			
Policy: modify	x					
Query: view	x		x	x	x	x

	ITM SuperUser	ITM Admin	Console Operator	Report User	Application Admin	System Admin
Query: modify	x			x		
Situation: view	x		x		x	x
Situation: stop/start	x		x		x	x
Situation: modify	x					
Terminal script: view	x		x		x	x
Terminal script: modify	x					
User Admin: modify	x	x				
User Admin: view	x	x				
User Admin: author mode eligible	x	x	x		x	x
User Admin: administration mode eligible	x					
Workspace Administration: workspace admin mode	x					
Workspace Administration: workspace author mode	x		x		x	x

Backing up user accounts and workspaces

It is important to back up the user accounts and workspaces. The information is stored in each TEPS database. You can use your normal database backup methodology.

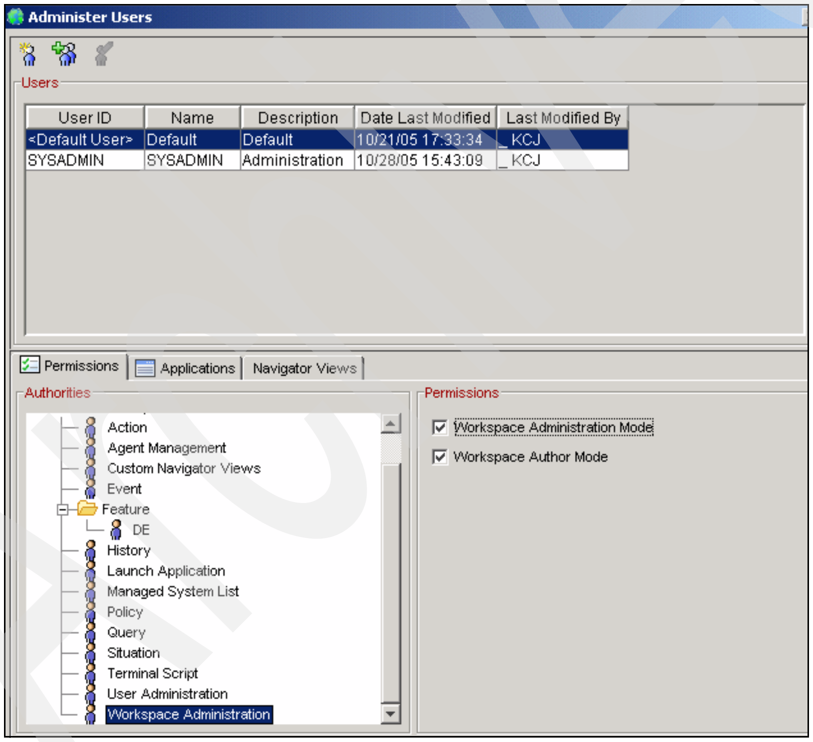
Workspace administration mode

An important part of the administration of users and workspaces is called Workspace Administration Mode. In order to protect workspaces, every user can only impact their own workspace (why everyone should have their own account) with the exception of when in workspace administration mode. In this mode, the administrator can publish workspaces as the default.

Tip: Workspaces, scripts, links, and so forth created while in workspace administration mode cannot be deleted *except* while in workspace administration mode. When making changes to navigators, you should ask all logged-on users to log off when you save your changes to the workspace.

The publishing of workspaces involves a five-step process:

1. Navigate to the workspace you wish to publish.
2. Open the Administer Users window (Ctrl+u) and on the Permissions tab, under authorities, select **Workspace Administration**.
3. The attribute for Workspace Administration mode should be unchecked except when you want to publish a workspace. Check it and close the Administer Users window (Figure 13-2).


Administer Users

Users

User ID	Name	Description	Date Last Modified	Last Modified By
<Default User>	Default	Default	10/21/05 17:33:34	_KCJ
SYSADMIN	SYSADMIN	Administration	10/28/05 15:43:09	_KCJ

Permissions | Applications | Navigator Views

Authorities
- Action
- Agent Management
- Custom Navigator Views
- Event
- Feature
  - DE
- History
- Launch Application
- Managed System List
- Policy
- Query
- Situation
- Terminal Script
- User Administration
- Workspace Administration

Permissions
- [✓] Workspace Administration Mode
- [✓] Workspace Author Mode

Figure 13-2 Activating workspace administration mode

4. Save the workspace. This has the effect of publishing it to all users who are using this workspace. The effects are nearly immediate to all logged-on users.

5. Repeat steps 2 and 3 and uncheck the Workspace Administration mode this time. Close the window.

Process authentication for take action commands

For architects and administrators who are familiar with the concept of running a process in the context of a different user, we now explain the Take Action feature and how it uses user context.

Important: The default product behavior is to run the action at the agent with the authority of the user the agent runs as. There is no checking done to see whether the person who initiates the activity has the right to issue the command by default.

User ID information is sent with the command instructions to the agent, but the default behavior does not use this information. The user ID sent varies from facility to facility within IBM Tivoli Monitoring 6.1. Table 13-2 outlines the type of information being passed.

Table 13-2 Type of information being passed

Type of action	**User ID type being passed**
Take action from event	User ID of logged-on user
Situation action	User ID of person who last updated situation
Workflow action	User ID of person who last updated policy

To cause the managed system to evaluate the information that it is passing, include the command prefix `security exit`. The limitation of this is that the command is still executed in the context of the user the agent runs as. If you use a security exit, you must define and maintain separate definitions for each agent type on each platform. Of course, in heterogeneous environments this can still be quite troublesome.

On UNIX, you can use the setuid method and cause the agent to switch to a different user context prior to execution of the command and then cause it to switch back to the correct user.

13.1.4 Sundry features

The following are sundry features of IBM Tivoli Monitoring 6.1.

Sound

You can assign alert sounds to your console when a situation changes from false to true. In some small environments this might have its use, in particular to demonstrate in a classroom setting that a situation has changed, bit this is not likely to be the normal experience.

In most environments, sound is not an effective tool and it should not be relied on heavily. There are frequently distractors in the environment that might cause the operator to not register the sound in their mind. It is also possible that your environment might have so many sounds playing on the workstation that the operator might be tempted to mute the audio. For these reasons, it is not recommended that you build a process around relying on sound to cue an operator. It is easy to build sufficiently attentive environments without relying on sound effects.

Shift

You can use a function in situation monitoring (and account for it with Historical Data Collection) for shift monitoring. This means monitoring for situations only during production hours. Be sure that you understand what this functionality does and how it works, particularly for multiple time-zone environments to avoid unintended outages and lack of awareness of those.

Restriction: The use of the shift functionality would seem to preclude the usability of a tool such as the Tivoli Service Level Adviser unless service levels are written around shifts instead of real time. When SLA is integrated more closely with ITM 6.x in a future release of SLA, we will see how this will be handled.

13.1.5 Monitoring situations

The following monitoring situations are features of IBM Tivoli Monitoring 6.1.

Configuration

Situations that run at an agent have a sampling interval. Situations that run at the TEMS also have a sampling interval. When you are watching the event console in the TEP, you should be aware that your refresh rate for the desktop client (or Web client) has to pass before you see the change in the situation status. Do not set the refresh rate to refresh too quickly, as this can overwhelm the TEPS with requests when there are many users.

Expert advice

Best practices for your in-house created situations is to provide as much expert advice as possible. Because the advice is sent from the TEPS server, be aware of the volume of data that will be required to be handled by the TEPS server (for example, if you choose to include .mpg files and such materials of sufficient size to bottleneck the server if many users might be pulling the information simultaneously).

13.1.6 Test/QA environments

It is very important to have a set of adequate test and QA environments. These environments do not have to be exactly the same as the production environment, but they should be identical to each other at least. Nothing should exist in the production environment that does not exist in the test and QA environments.

In the test and QA environments, users should have permissions exactly as they should in the production environment by roles. This enables the implementation of process and product changes in the production environment with a high degree of confidence on impact.

The QA and test environments should include systems from the test and QA environments of monitored applications in the environment.

13.1.7 Command-line functionality

At this time, the functionality of the command line is limited to basic product functions such as configuration, agent management, and some troubleshooting. There is also a SOAP interface, although it is generally not recommended at this time to build extensively on it. The ITM Admin Guide covers some of the more useful and likely to be used SOAP methods and tags with examples of usage.

IBM provides numerous SOAP methods with IBM Tivoli Monitoring Web services. These methods enable you to dynamically query and control IBM Tivoli Monitoring environments. Using IBM SOAP methods, you can:

- Stop or start policies and situations.
- Forward AF/Remote trapped messages and display them on a Universal Message console.
- Retrieve attribute data that you can display in charts or reports.
- Open and close events.
- Make real-time requests for data.
- Issue SOAP requests as system commands in Tivoli Enterprise Portal.

You can also use this product to test a request to ensure that it works properly. You can then create a policy that submits multiple requests for processing. You can also generate daily operation summaries. You can store retrieved data in the Tivoli Data Warehouse, as described in the Historical Data Collection guide.

Note: IBM Tivoli Monitoring Web Services provides XML data rows. Use IBM SOAP methods in combination with your own scripts to display the data in charts and tables.

Important command line functions you should be familiar with include:

tacmd addBundles	Adds the bundle files to the depot for deployment.
tacmd addSystem	Deploys an agent and required components to a managed system.
tacmd createNode	Deploys the OS agent to a new system.
tacmd listSystems	Lists the agents running on a particular system and their status.
tacmd login	Login is required prior to being able to issue any other commands.
tacmd logout	Logs out the user prior to the default logout of 15 minutes.
tacmd restartAgent	Cycles the agent on a managed system.
tacmd startAgent	Starts the agent on a managed system.
tacmd stopAgent	Stops the agent on a managed system.
tacmd updateAgent	Used when upgrading agent components.
tacmd viewAgent	Provides detailed information about an agent on a system.

In 8.2.8, "Create, test, and implement the monitor" on page 480, we discussed using `tacmd viewSit` and `createSit` to export and import situations.

In addition to the aforementioned selected commands, there are other very important commands, which are available only to UNIX, that you should be familiar with if you have UNIX systems:

cinfo Has a menu of sub-commands such as:
-- CINFO Menu --
1) Show products installed in this CandleHome
2) Show which products are currently running
3) Show configuration settings
4) Show installed CD release versions
X) Exit CINFO

itmcmd agent	Use = to start and stop most agents. You can start or stop one agent, all agents, or multiple agents.
itmcmd manage	Use to start Manage Tivoli Enterprise Monitoring Services on a UNIX or Linux computer. You can start, stop, and configure monitoring components in Manage Tivoli Enterprise Monitoring Services.
itmcmd server	Use to start and stop monitoring servers that are defined in directories under the $itmcmd server/tables subdirectory. You must run this command from the host computer.

There are also many other scripts that you should know about, including:

addrbfile.p	Add the new rb files for TEC integration manually to the TEC rulebase.
itmtec-remove.sh	Uninstall of ITM-TEC synchronization tool.
query_state.cmd	Query state of the Situation update forwarder (SUF) process.
startSUF.cmd	Start SUF process manually, to be run in background (&).
stop.cmd	Stop SUF process.
test.cmd	Test connection of SUF process to the defined TEMS servers.
upg_sentry_baroc.pl	Upgrades only the sentry BAROC files manually.
wcrtitmtask.sh	Creates the new ITM synchronization tasks manually.

13.1.8 Workspace administration

One topic that was not covered in Chapter 8, “Real-life scenarios” on page 469 is links. You can create links to provide shortcuts or navigation to workspaces or navigator views. Recall our example in 8.4, “Building another example: monitoring Tivoli” on page 499, as an example of a custom workspace for the Tivoli Administrator where we can build a link:

1. Returning to our Framework workspace (see Figure 13-3 on page 713) we decide to create a link from the madrid icon on the workspace to the ITM administrator workspace.

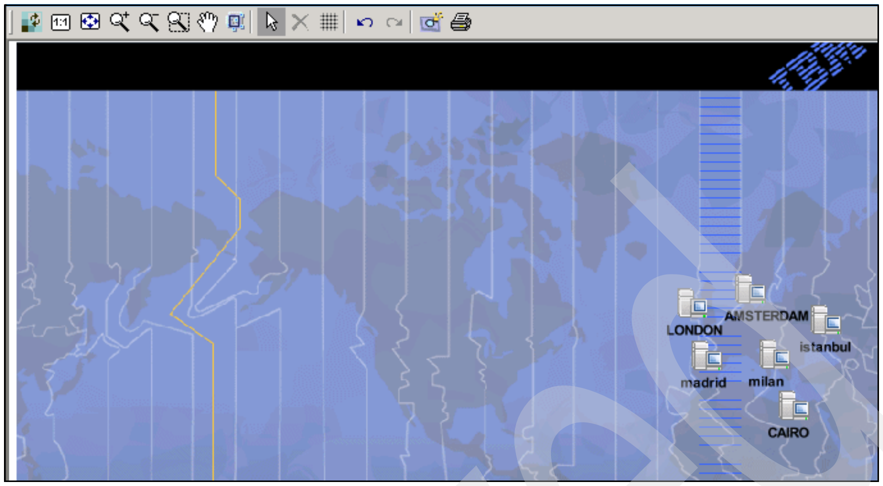


Figure 13-3 Our Framework administrator view

2. Choose the arrow tool and highlight the **madrid** icon. Right-click the icon and select **Link to** → **Link wizard**. Use the wizard to create a link to the workspace that is to be attached. When the link is completed, a small chain link has been added to the icon, as shown in Figure 13-4. (This is optional; right-click and deselect it if you do not want to have it visible.)

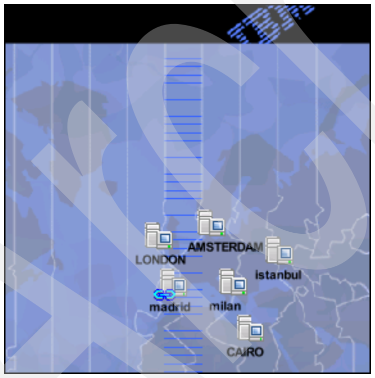


Figure 13-4 Madrid with a link assigned to it

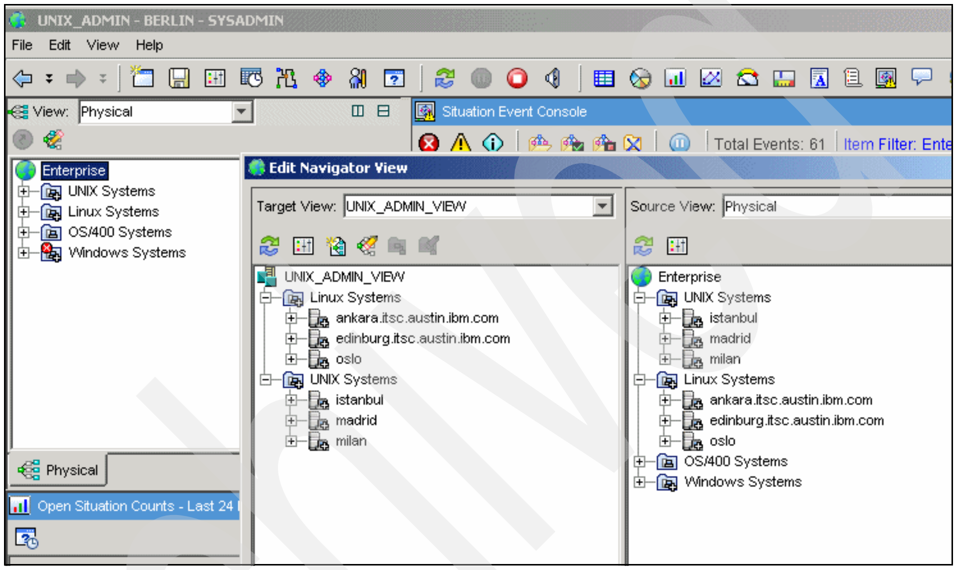
3. When you are using the hand (normal mode) on the workspace, clicking **madrid** takes you to the selected workspace view.

A more interesting option in links is relative links. What we just created was an absolute link. Relative links enable you to create workspaces that are linked in the context of some information being passed through selection of the link.

Building a workspace for a console operator

As was previously mentioned, you will likely want to restrict the access that a user has to a certain group of machines as far as events they can see and manage. First, we build a workspace and restrict it to some machines:

1. Create a new workspace by saving a copy of it. Create a new Navigator item. Add the physical trees for Linux and UNIX to the tree. The resulting logical structure should look like Figure 13-5.



View: Physical

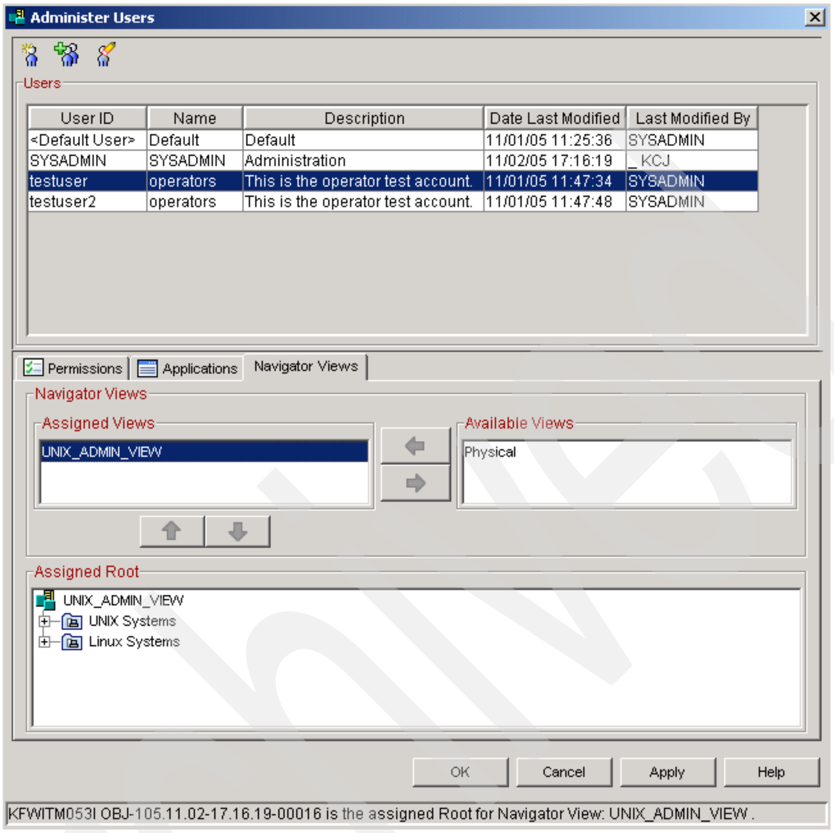
Target View: UNIX_ADMIN_VIEW

Source View: Physical

Figure 13-5 Creating a Linux/UNIX Admin workspace

2. Publish the workspace: Go into workspace author mode (see "Workspace administration mode" on page 706 for instructions to do that) and save the workspace.

3. Assign the workspace to a user as shown in Figure 13-6.



User ID	Name	Description	Date Last Modified	Last Modified By
<Default User>	Default	Default	11/01/05 11:25:36	SYSADMIN
SYSADMIN	SYSADMIN	Administration	11/02/05 17:16:19	_KCJ
testuser	operators	This is the operator test account.	11/01/05 11:47:34	SYSADMIN
testuser2	operators	This is the operator test account.	11/01/05 11:47:48	SYSADMIN

Assigned Views: UNIX_ADMIN_VIEW

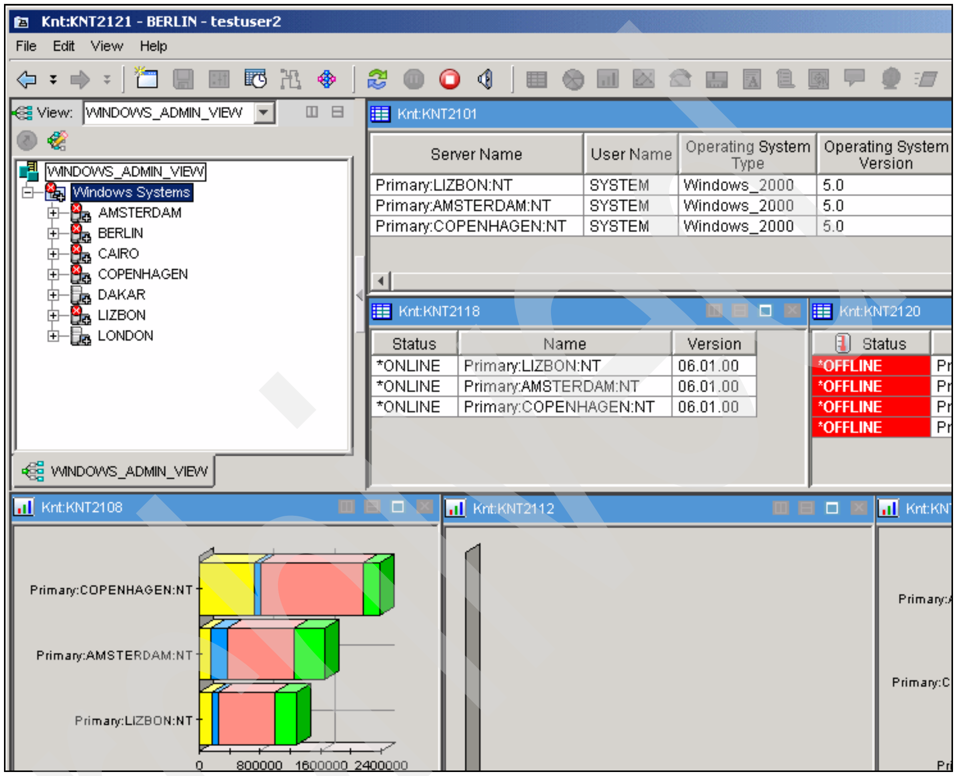
Available Views: Physical

Assigned Root:
- UNIX_ADMIN_VIEW
  - UNIX Systems
  - Linux Systems

KFWITM053I OBJ-105.11.02-17.16.19-00016 is the assigned Root for Navigator View: UNIX_ADMIN_VIEW .

Figure 13-6 Assigning only the UNIX workspace to this user

4. Similarly, we created a workspace for Windows admins only and published it. Then assigned it to the second testuser as the only workspace. Figure 13-7 and Figure 13-8 on page 717 show the view for the Windows administrator and the UNIX admins respectively.

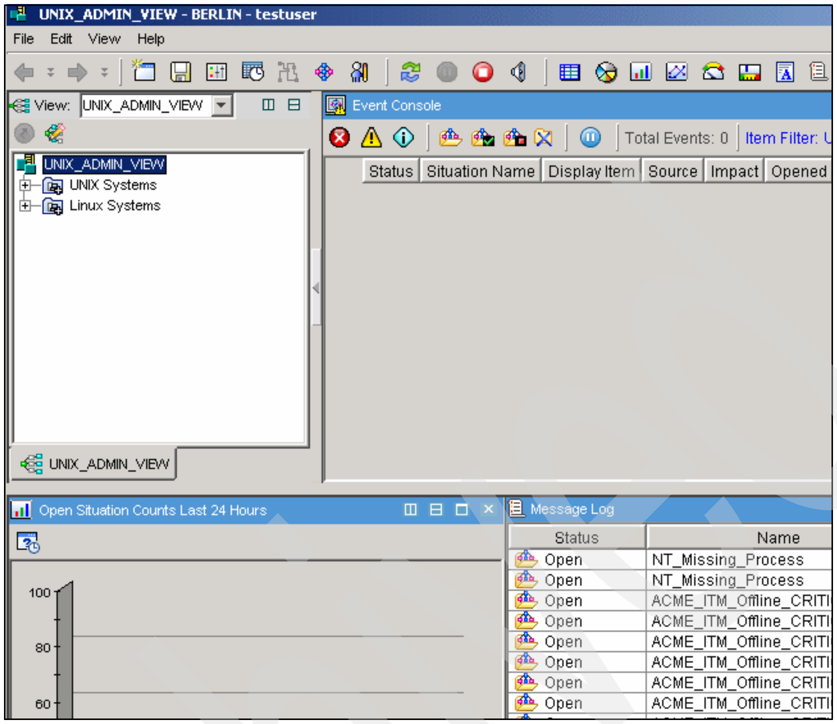


Server Name	User Name	Operating System Type	Operating System Version
Primary:LIZBON:NT	SYSTEM	Windows_2000	5.0
Primary:AMSTERDAM:NT	SYSTEM	Windows_2000	5.0
Primary:COPENHAGEN:NT	SYSTEM	Windows_2000	5.0

Status	Name	Version
*ONLINE	Primary:LIZBON:NT	06.01.00
*ONLINE	Primary:AMSTERDAM:NT	06.01.00
*ONLINE	Primary:COPENHAGEN:NT	06.01.00

Status	
*OFFLINE	Pr
*OFFLINE	Pr
*OFFLINE	Pr
*OFFLINE	Pr

Figure 13-7 The Windows admin workspace



Status	Situation Name	Display Item	Source	Impact	Opened

Status	Name
Open	NT_Missing_Process
Open	NT_Missing_Process
Open	ACME_ITM_Offline_CRITI
Open	ACME_ITM_Offline_CRITI
Open	ACME_ITM_Offline_CRITI
Open	ACME_ITM_Offline_CRITI
Open	ACME_ITM_Offline_CRITI
Open	ACME_ITM_Offline_CRITI

Figure 13-8 A view of our less eventful UNIX environment

You can assign fewer systems to the operator's environments if you wish (for example, as in the case of an SAP admin to whom you want to give a console view based on a single set of systems).

You might consider some variations in the assigned privileges based on your intent to the way your organization will use the tool. If you only want the console operator (not system admins) to acknowledge and close events, thus owning the issue until resolution, you will want to remove those privileges from those classes of users. However, if you want to have them track their progress and make comments in the acknowledgement box (the system admins) they require privileges to acknowledge the events, at least.

13.2 Maintenance topics

This section covers maintenance information, including validation, PMRs, logs, and processing.

13.2.1 Validating situations and agent status

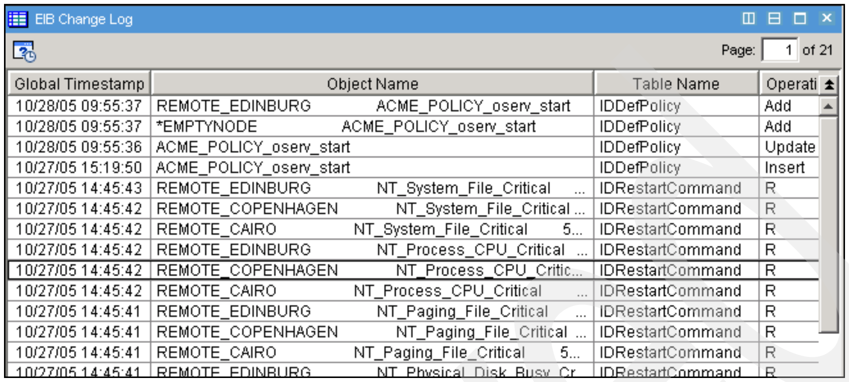
After you install something, there may be other users with the authority or technical security authorization to stop or remove components from machines in the environment. There is also a high probability that things will change in the environment, resulting in issues in IBM Tivoli Monitoring 6.1 infrastructure. The following points should help you to be well on your way to arriving at a strategy that your team and organization will use to deal with these types of issues.

- For new machines added to the environment, ideally, a change record should exist in change management for this, and your group (or the system administrators, whomever is responsible for new agents) should be triggered by this.
- Likewise, the removal of a machine from the environment should require a change record as well.
- That having been said, in most environments, the change process has so many holes that many systems slip in and out of the environment completely off the radar. In those types of situations there are several options:
  - Review the situation event console frequently to understand when situations are stopped. Investigate situations that are stopped frequently.
  - Review the managed system status console frequently to see the status of agents.
  - Use IBM Tivoli NetView to search for machines in the environment that are not running the IBM Tivoli Monitoring 6.1 Agent. This involves attempting to communicate with the system console on each machine (port 1920 by default). Review the lists of machines where this port does not respond.
  - Review incident and problem management reports looking for systems not managed by IBM Tivoli Monitoring 6.1.
  - Give presentations and make them available about the monitoring services that you provide. Others may report systems that they are aware are not monitored.

You should also be familiar with the following resources.

EIB Change Log

The EIB Change Log (Figure 13-9) records information about changes to situations, policies, user definitions, and configuration information.


EIB Change Log

Page: 1 of 21

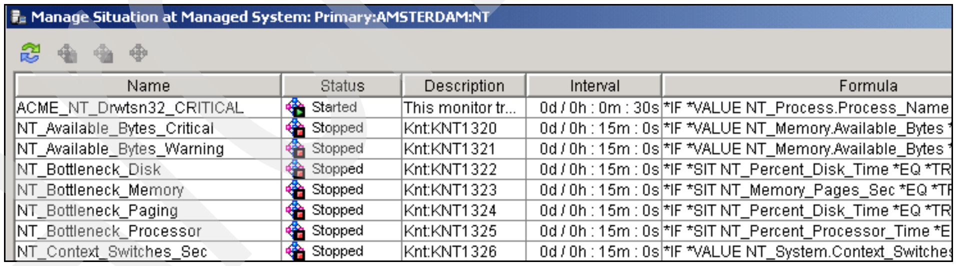
Global Timestamp	Object Name	Table Name	Operati
10/28/05 09:55:37	REMOTE_EDINBURG ACME_POLICY_oserv_start	IDDefPolicy	Add
10/28/05 09:55:37	*EMPTYNODE ACME_POLICY_oserv_start	IDDefPolicy	Add
10/28/05 09:55:36	ACME_POLICY_oserv_start	IDDefPolicy	Update
10/27/05 15:19:50	ACME_POLICY_oserv_start	IDDefPolicy	Insert
10/27/05 14:45:43	REMOTE_EDINBURG NT_System_File_Critical ...	IDRestartCommand	R
10/27/05 14:45:42	REMOTE_COPENHAGEN NT_System_File_Critical ...	IDRestartCommand	R
10/27/05 14:45:42	REMOTE_CAIRO NT_System_File_Critical 5...	IDRestartCommand	R
10/27/05 14:45:42	REMOTE_EDINBURG NT_Process_CPU_Critical ...	IDRestartCommand	R
10/27/05 14:45:42	REMOTE_COPENHAGEN NT_Process_CPU_Critic...	IDRestartCommand	R
10/27/05 14:45:42	REMOTE_CAIRO NT_Process_CPU_Critical ...	IDRestartCommand	R
10/27/05 14:45:41	REMOTE_EDINBURG NT_Paging_File_Critical ...	IDRestartCommand	R
10/27/05 14:45:41	REMOTE_COPENHAGEN NT_Paging_File_Critical ...	IDRestartCommand	R
10/27/05 14:45:41	REMOTE_CAIRO NT_Paging_File_Critical 5...	IDRestartCommand	R
10/27/05 14:45:41	REMOTE_EDINBURG NT_Physical_Disk_Busy_Cr	IDRestartCommand	R

Figure 13-9 The EIB Change Log

Start at the Enterprise Navigator, select **Workspaces** from the menu, and choose **EIB Change Log** as the workspace.

Managed Situations Editor

When selecting a type of agent from the Navigator, right-click and select **Managed Situations.** This view (Figure 13-10) shows the settings for all situations running on this agent. It also allows for configuration of these items.

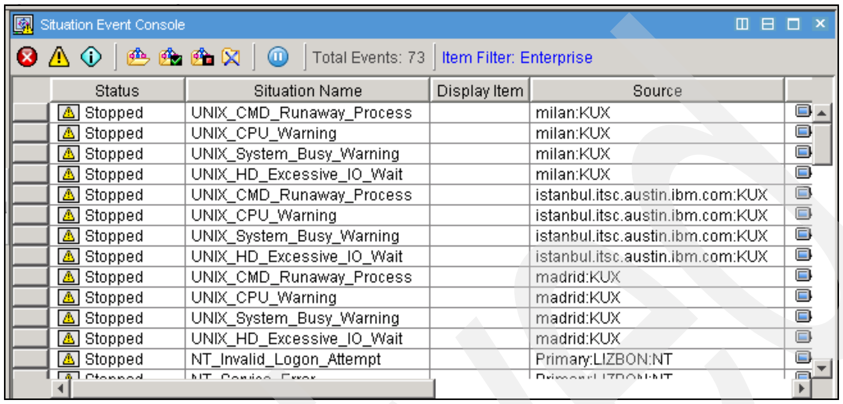

Manage Situation at Managed System: Primary:AMSTERDAM:NT

Name	Status	Description	Interval	Formula
ACME_NT_Drwtsn32_CRITICAL	Started	This monitor tr...	0d / 0h : 0m : 30s	*IF *VALUE NT_Process.Process_Name
NT_Available_Bytes_Critical	Stopped	Knt:KNT1320	0d / 0h : 15m : 0s	*IF *VALUE NT_Memory.Available_Bytes
NT_Available_Bytes_Warning	Stopped	Knt:KNT1321	0d / 0h : 15m : 0s	*IF *VALUE NT_Memory.Available_Bytes
NT_Bottleneck_Disk	Stopped	Knt:KNT1322	0d / 0h : 15m : 0s	*IF *SIT NT_Percent_Disk_Time *EQ *TR
NT_Bottleneck_Memory	Stopped	Knt:KNT1323	0d / 0h : 15m : 0s	*IF *SIT NT_Memory_Pages_Sec *EQ *TR
NT_Bottleneck_Paging	Stopped	Knt:KNT1324	0d / 0h : 15m : 0s	*IF *SIT NT_Percent_Disk_Time *EQ *TR
NT_Bottleneck_Processor	Stopped	Knt:KNT1325	0d / 0h : 15m : 0s	*IF *SIT NT_Percent_Processor_Time *E
NT_Context_Switches_Sec	Stopped	Knt:KNT1326	0d / 0h : 15m : 0s	*IF *VALUE NT_System.Context_Switches

Figure 13-10 Managed Situations Editor

Situation Event Console

The Situation Event Console (Figure 13-11) details the current status of the agents and situations within the purview of the user viewing them. It is also a place to start and stop situations if user permissions allow.



Situation Event Console

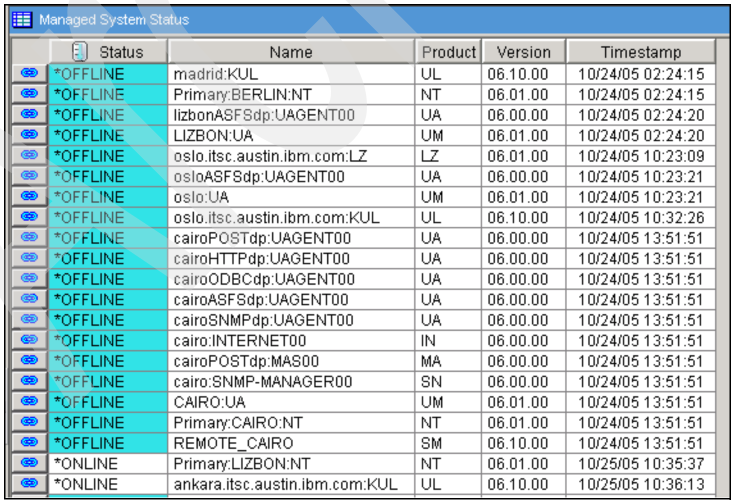
Total Events: 73 Item Filter: Enterprise

Status	Situation Name	Display Item	Source
⚠ Stopped	UNIX_CMD_Runaway_Process		milan:KUX
⚠ Stopped	UNIX_CPU_Warning		milan:KUX
⚠ Stopped	UNIX_System_Busy_Warning		milan:KUX
⚠ Stopped	UNIX_HD_Excessive_IO_Wait		milan:KUX
⚠ Stopped	UNIX_CMD_Runaway_Process		istanbul.itsc.austin.ibm.com:KUX
⚠ Stopped	UNIX_CPU_Warning		istanbul.itsc.austin.ibm.com:KUX
⚠ Stopped	UNIX_System_Busy_Warning		istanbul.itsc.austin.ibm.com:KUX
⚠ Stopped	UNIX_HD_Excessive_IO_Wait		istanbul.itsc.austin.ibm.com:KUX
⚠ Stopped	UNIX_CMD_Runaway_Process		madrid:KUX
⚠ Stopped	UNIX_CPU_Warning		madrid:KUX
⚠ Stopped	UNIX_System_Busy_Warning		madrid:KUX
⚠ Stopped	UNIX_HD_Excessive_IO_Wait		madrid:KUX
⚠ Stopped	NT_Invalid_Logon_Attempt		Primary:LIZBON:NT

Figure 13-11 Situation Event Console

Managed System Status view

From the Enterprise Navigator, right-click and select **Workspaces** → **Managed System Status** view. From this view you can see the status of all agents for all managed systems in one clean view (Figure 13-12). It also shows version numbers and the timestamp at which this status was last current. You need to familiarize yourself with the product codes to understand this view.



Managed System Status

Status	Name	Product	Version	Timestamp
*OFFLINE	madrid:KUL	UL	06.10.00	10/24/05 02:24:15
*OFFLINE	Primary:BERLIN:NT	NT	06.01.00	10/24/05 02:24:15
*OFFLINE	lizbonASFSdp:UAGENT00	UA	06.00.00	10/24/05 02:24:20
*OFFLINE	LIZBON:UA	UM	06.01.00	10/24/05 02:24:20
*OFFLINE	oslo.itsc.austin.ibm.com:LZ	LZ	06.01.00	10/24/05 10:23:09
*OFFLINE	osloASFSdp:UAGENT00	UA	06.00.00	10/24/05 10:23:21
*OFFLINE	oslo:UA	UM	06.01.00	10/24/05 10:23:21
*OFFLINE	oslo.itsc.austin.ibm.com:KUL	UL	06.10.00	10/24/05 10:32:26
*OFFLINE	cairoPOSTdp:UAGENT00	UA	06.00.00	10/24/05 13:51:51
*OFFLINE	cairoHTTPdp:UAGENT00	UA	06.00.00	10/24/05 13:51:51
*OFFLINE	cairoODBCdp:UAGENT00	UA	06.00.00	10/24/05 13:51:51
*OFFLINE	cairoASFSdp:UAGENT00	UA	06.00.00	10/24/05 13:51:51
*OFFLINE	cairoSNMPdp:UAGENT00	UA	06.00.00	10/24/05 13:51:51
*OFFLINE	cairo:INTERNET00	IN	06.00.00	10/24/05 13:51:51
*OFFLINE	cairoPOSTdp:MAS00	MA	06.00.00	10/24/05 13:51:51
*OFFLINE	cairo:SNMP-MANAGER00	SN	06.00.00	10/24/05 13:51:51
*OFFLINE	CAIRO:UA	UM	06.01.00	10/24/05 13:51:51
*OFFLINE	Primary:CAIRO:NT	NT	06.01.00	10/24/05 13:51:51
*OFFLINE	REMOTE_CAIRO	SM	06.10.00	10/24/05 13:51:51
*ONLINE	Primary:LIZBON:NT	NT	06.01.00	10/25/05 10:35:37
*ONLINE	ankara.itsc.austin.ibm.com:KUL	UL	06.10.00	10/25/05 10:36:13

Figure 13-12 The Managed System Status workspace

13.2.2 Opening a PMR with IBM

For support for issues with IBM Tivoli Monitoring 6.1, the following information will be helpful. You should prepare these requests prior to placing your call so you can be ready to submit the information to IBM Tivoli Software L2 support.

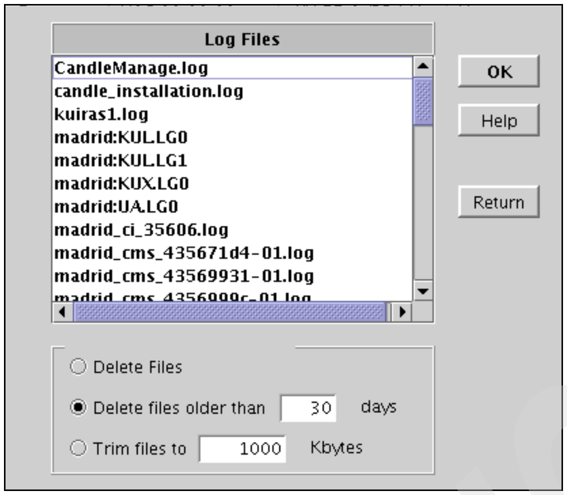
- The Windows version and service pack on the TEP Client and TEPS Server
- TEMS version and platform (determine in Manage Tivoli Enterprise Monitoring Services)
- Running TEP in desktop or browser mode
- If browser mode: IE version at client
- Agent types, versions, and where deployed
- All necessary environmental information including the version of TEP and the build level (from the TEP Client kcjras1.log file)
- All logs from all components involved
- TEMS log kmsras1: always necessary, may include additional information

13.2.3 Logs

Be aware of certain limitations surrounding log files:

- The TEP message log is purged when the TEP desktop starts.
- The message log at TEMS is erased when the TEMS is cycled.

Careful monitoring of file systems and drives is suggested as the logs are not rotated automatically. The UNIX MTEMS has a facility for some management of log files, as shown in Figure 13-13 on page 722.



Log Files

```
CandleManage.log
candle_installation.log
kuiras1.log
madrid:KUL.LG0
madrid:KUL.LG1
madrid:KUX.LG0
madrid:UA.LG0
madrid_ci_35606.log
madrid_cms_435671d4-01.log
madrid_cms_43569931-01.log
```

( ) Delete Files
(●) Delete files older than 30 days
( ) Trim files to 1000 Kbytes

Figure 13-13 UNIX MTEMS log file facility

This facility gives some options to prune the logs, delete files, or delete files older than a certain number of days. There is no equivalent in the Windows MTEMS at this time. There is also a CLI equivalent for managing logs for UNIX.

It is also helpful to know where the logs are stored. This is subject to change in the GA version of the product.

TEP Server:

```
\IBM\ITM\logs\install_dir\logs\hostname_cq_HEXtimestamp-nn.log
```

TEP Desktop Client:

```
\IBM\ITM\CNP\logs\kcjras1.log
```

TEP Browser Client:

```
C:\Documents and Settings\Administrator\Application
Data\IBM\Java\Deployment\log\plugin142.trace
```

TEMS:

```
\IBM\ITM\logs\install_dir\logs\hostname_ms_HEXtimestamp-nn.log
\IBM\ITM\logs\install_dir\logs\hostname_cms_HEXtimestamp-nn.log
\IBM\ITM\cms\cms.msg
```

TEMAs:

```
\IBM\ITM\logs\install_dir\logs\hostname_PC_HEXtimestamp-nn.log
IBM\ITM\TMAITM6\logs\*..lg0
```

(*PC* stands for product code.)

13.2.4 Process information

The following information details the process information to be able to identify the running ITM processes:

- Windows

TEP browser client	iexplore.exe
TEP desktop (Args: …Dcnp.http.url.host=<hostname> -Dvbroker.agent.enableLocator=false candle.fw.pres.CMWApplet)	clientjava.exe
TEP Server	KfwServices.EXE
TEMS	kdsmain.exe and cms.exe
Windows Agent	kntcma.exe
TE Universal Agent	kuma610.exe
Other Agents	<pc>*.exe – pc=product code
Tivoli Enterprise Monitoring Services	kinconfg.exe

- UNIX

TEMS	kdsmain and cms
UNIX OS Agent	kuxagent
UNIX log Agent	kulagent

14

Troubleshooting

This chapter introduces the user to the troubleshooting components of the IBM Tivoli Monitoring 6.1 product set. This product set is quite complex and configured and this means that administrators of this product have to be aware of the environment and have the ability to troubleshoot and correct issues that arise. This chapter gives some insight into the steps that can be used to troubleshoot and correct problems with the product. It is not exhaustive, and in some cases IBM Support will be required to isolate issues and correct them.

In this chapter, we cover the following topics:

- Message logging
- Trace facilities
- Using the product documentation
- Tools
- Sample problem scenarios

14.1 Overview

With a complex and configurable product like IBM Tivoli Monitoring 6.1 there is a need for troubleshooting skills to deploy and maintain the environment. Over the lifecycle of using this product issues will arise; some can be addressed by administrators and others require assistance from IBM Support. This chapter provides guidelines to allow for efficient troubleshooting of the IBM Tivoli Monitoring 6.1 product.

Troubleshooting can be defined as the process of identifying, isolating, and ultimately resolving problems. In the case of IBM Tivoli Monitoring 6.1, several steps can be used to go through the troubleshooting process.

Identifying the problem is the first step where the product's error messages and trace logs can be critical. Second is the potential re-creation of the problem with increased trace settings depending on the issue. Then, there is the use of the Problem Determination guides and other tools to aid in the resolution of the problem.

This chapter covers many aspects of the product; however, the product user's guides and administration guides provide invaluable information that can be used in troubleshooting. Additionally chapters in this book also contain some troubleshooting information that makes more sense in their original chapters, so they are not repeated here.

14.2 Message logging

Each component of IBM Tivoli Monitoring 6.1 from the servers (TEMS/TEPS) to the agents has messaging facilities to provide feedback about exceptional issues that have occurred. The messages can be informational, warning, or error in nature depending on the message. Message output destination can be the screen or a log file. These messages are documented in *IBM Tivoli Monitoring, Version 6.1.0, Problem Determination Guide,* GC32-9458.

These are some samples of the three types of messages:

Informational	`KFWITM006I Validating user credentials`
Warning	`KFWITM197W User has no assigned Navigator Views`
Error	`KFWITM215E Unable to process logon request`

14.3 Trace facilities

IBM Tivoli Monitoring 6.1 contains an extensive trace facility that can provide helpful information about the state of the components. Several types of logs are created by IBM Tivoli Monitoring 6.1, with the principal log type being the reliability, availability, and serviceability (RAS). RAS logs are in English and are available on the Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, and monitoring agents. Other logs include installation, seed, LG0, ODBC, and other configuration files. In this section, we cover some of these trace settings and how to read the trace logs that are generated.

Table 14-1 summarizes the IBM Tivoli Monitoring 6.1 log names and locations.

Table 14-1 Log names and locations

Windows		**UNIX**	
TEPS	*install_dir*\logs\hostname_cq_timestamp-xx.log	TEPS - Linux	*install_dir*/logs/*hostname_cq_timestamp-xx*.log
TEMS	*install_dir*\logs\hostname_ms_*timestamp*-xx.log	TEMS	*install_dir*/logs/*hostname_ms_timestamp-xx*.log
Agents	*install_dir*\tmaitm6\logs Log names vary by agent. ► RAS1 logs generally have the syntax of *hostname_PC_timestamp-xx*.log ► The *.LG0 log file shows the connectivity with the TEMS, situations running, and status of Take Actions.	Agents	install_dir/logs Log names vary by agent. ► RAS1 logs generally have the syntax of *hostname_PC_timestamp-xx*.log ► The *.LG0 log file shows the connectivity with the TEMS, situations running, and status of Take Actions.
Warehouse Proxy	*install_dir*/logs/*hostname*_hd_*timestamp*-xx.log	Warehouse Proxy	Currently not available on UNIX
tacmd	*install_dir*\bin\kuiras1.log	tacmd	*install_dir*/logs/kuiras1.log
ITM 5.x Monitoring Agent	Endpoint logs: %LCF_DATDIR%/LCFNEW/KTM/logs/ktmras1.log TMR logs: %DBDIR%/KTM/logs ► trace_witm61agt___#_pid.log ► trace_ITM61Integration_Install____#_pid.log	ITM 5.x Monitoring Agent	Endpoint logs: $LCF_DATDIR/LCFNEW/KTM/logs/ktmras1.log TMR logs: $DBDIR/KTM/logs ► trace_witm61agt___#_pid.log ► trace_ITM61Integration_Install_____#_pid.log

Windows		UNIX	
Seeding process	*install_dir*\CNPS\logs\seedPPC.log *install_dir* InstallITM\logs\CMSSeed.log for the TEMS	Seeding process	*install_dir*/logs/*hostname*_ci_<*tems pid*>.log
Summarization and Pruning agent	*install_dir*/logs/*hostname_sy_timestamp-xx*.log	Summarization and Pruning agent	*install_dir*/logs/*hostname_sy_timestamp-xx*.log
Installation logs	*install_dir*\InstallITM ► Main installation log: IBM Tivoli Monitoring *date PID*.log ► Configuration of the TEPS ODBC connection: TEPS_ODBC.log ► Configuration of the Warehouse proxy: Warehouse_Configuration.log	Installation logs	install_dir/logs ► Main installation log: candle_installation.log ► TEPS seeding log: InstallPresentation.log ► TEPS configuration log (assuming DB2 is the database): db2prep.log

Note: In Table 14-1, the following variables are defined:

xx	The rotating log number
PC	The two-letter product code (for example, UX for the UNIX Agent)
PPC	The three-letter product code (for example, KUX for the UNIX Agent)

14.3.1 Trace settings

Tracing of components is controlled by several environment variables, and there are several methods by which these variables can be modified. Table 14-2 defines many of the environment variables. (This is not an exhaustive list.)

Table 14-2 Trace environment variables

Variable	Description
KBB_RAS1	Controls the trace level in the RAS logs.
KDC_DEBUG	Diagnosing communications and connectivity problems.
KBB_RAS1_LOG	Logfile location of the RAS1 log.
INVENTORY	File containing the inventory of RAS1 logs for the component.

Variable	Description
MAXFILES	Total number of log files to maintain. Default is 32 MB.
LIMIT	Maximum log file size per file in MB. Default is 5.
COUNT	Maximum number of log files per session. Default is 5.
Universal Agent specific settings	
KUMP_ODBC_DEBUG=Y	ODBC Data Provider tracing.
KUMP_HTTP_DEBUG=Y	HTTP Data Provider tracing.
KUMP_SCRIPT_DEBUG=Y	Script Data Provider tracing.
KUMP_SNMP_DEBUG_TRAP=Y KUMP_SNMP_DEBUG_DISCOVERY_ROUTE=Y KUMP_SNMP_DEBUG_DISCOVERY_NETWORK=Y KUMP_SNMP_DEBUG_MIB_MANAGER=Y KUMP_SNMP_DEBUG_MIB_IO=Y	SNMP Data Provider tracing. All of the debug environment variables listed above default to No. As an example, if you use the SNMP Data Provider and have problems collecting MIB data, you set these two environment variables: KUMP_SNMP_DEBUG_MIB_MANAGER=Y KUMP_SNMP_DEBUG_MIB_IO=Y
ERROR (UNIT:kumpfile Error State Detail Flow Metrics) (UNIT:kumpdcmf ALL)	Detailed File Data Provider tracing.
ERROR (UNIT:kumpsosr ALL) (UNIT:kumpspst ALL) (UNIT:kumpscku ALL) (UNIT:kumpstcp ALL) (UNIT:kumplpba ALL)	Detailed API or Socket Data Provider tracing.
ERROR (UNIT:kumamain ALL)	Problems involving managed system online/offline processing.
ERROR (UNIT:kumpdpda Error Output) (UNIT:kumpmd2a Error Detail)	Incorrect report data.
ERROR METRICS	Problems involving Universal Agent memory usage.

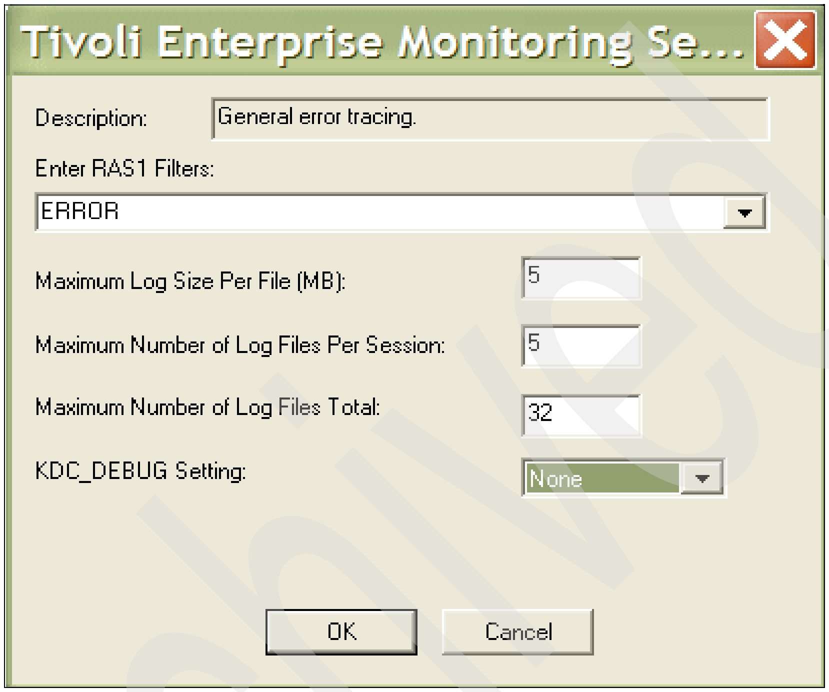
Environment variables can be modified using any of the following methods:

- Modify the appropriate environment file:
  - TEMS
    - Windows: install_dir\CMS\KBBENV
    - UNIX: install_dir/config/hostname_ms_hostname.config
  - TEPS
    - Windows: install_dir\CNPS\KFWENV
    - Linux: install_dir/config/cq.config
  - Agents
    - Windows: install_dir\TMAITM6\PPCENV (Where *PPC* is the three letter product code for the agent.)

      For example: Windows Agent: C:\IBM\ITM\TMAITM6\KNTENV
    - UNIX: install_dir/config/*pc*.config (Where *pc* is the two letter product code for the agent.)

      For example: Linux Agent: /opt/IBM/ITM/config/lz.config
  - Command line: **`tacmd`**
    - Windows: install_dir\bin\KUIENV
    - UNIX: install_dir\bin\tacmd
  - ITM 5.x Monitoring Agent
    - Windows: %LCF_DATDIR%\LCFNEW\KTM\KTMENV
    - UNIX: $LCF_DATDIR/LCFNEW/KTM/KTMENV

- Modify the trace settings using the Manage Tivoli Enterprise Monitoring Services GUI. Right-click the desired component and select **Advanced** → **Edit Trace Parms**. This displays a menu in which you can modify the trace settings (Figure 14-1).



Description: General error tracing.

Enter RAS1 Filters: ERROR

Maximum Log Size Per File (MB): 5

Maximum Number of Log Files Per Session: 5

Maximum Number of Log Files Total: 32

KDC_DEBUG Setting: None

Figure 14-1 Trace parameter menu for the TEMS

- Connect to the IBM Tivoli Monitoring Service Index using a browser:
  a. Use a browser to access:

     ```
     http://systemname:1920
     ```

  b. If multiple components are installed, select the appropriate one and enter a valid user and password for authentication.

     This displays the IBM Tivoli Monitoring Service Console for the selected component. At the bottom of the page, you can change settings. See *IBM Tivoli Monitoring Problem Determination Guide,* GC32-9458, for more details about using and blocking this tool.

After the trace settings are modified, recycle the corresponding component for the change to take affect. The log file for the component will show the current trace level in the header as seen in Example 14-1 on page 727.

Example 14-1 Header from TEMS log file

```
!435BFB0A.0000!======================>  IBM Tivoli RAS1 Service Log  <========================
+435BFB0A.0000      System Name: aixurania                   Process ID: 8344
+435BFB0A.0000     Program Name: kdsmain                      User Name: root
+435BFB0A.0000        Task Name: cms                        System Type: AIX;5.2
+435BFB0A.0000   MAC1_ENV Macro: 0xA326                      Start Date: 2005/10/23
+435BFB0A.0000       Start Time: 17:05:14                      AS Limit: None
+435BFB0A.0000       Core Limit: 1024M                        CPU Limit: None
+435BFB0A.0000       Data Limit: 2048M                      Fsize Limit: 1024M
+435BFB0A.0000     Nofile Limit: 2000                       Stack Limit: 32M
+435BFB0A.0000    Service Point: root.aixurania_ms       UTC Start Time: 435bfb0a
+435BFB0A.0000  Executable Name: kdsmain                       ITM Home: /opt/IBM/ITM
+435BFB0A.0000      ITM Process: aixurania_ms
+435BFB0A.0000         KBB_RAS1: ERROR (UNIT:kglcry all)
+435BFB0A.0000      KBB_ENVPATH: KBBENV
+435BFB0A.0000 ==============================================================================
```

14.3.2 Using the trace logs

Because there will most likely be several trace logs on a given system, it is advisable to know when the error occurred so that the correct log file can be accessed. After the correct log file is collected it can be viewed using any text editor or word processing program. If one of these programs is used to view the log, then the hexadecimal timestamp will not be converted; however, if the TMS Log Viewer is used to view the logs then this timestamp will be converted to human-readable format. The TMS Log Viewer can be accessed through the Manage Tivoli Enterprise Monitoring Services.

Knowing the type of issue being isolated can be helpful as well because the logs can be very verbose when trace levels are increased. For example, if the problem is a TEPS logon failure, then you could look at the TEPS log for the user ID entered when the failure occurred. (Note the response `Invalid Userid` in Example 14-2.)

Example 14-2 Example trace output

```
(43503C88.0000-1228:ctauthorizationevaluator_i.cpp,727,"CTAuthorization::Evalua
tor_i::executeQuery") Invalid Userid <operUser>
```

Above the trace level was the default of ERROR and the user did not exist in the TEPS database. In Example 14-3 on page 733, the trace level is increased and the user does exist in the TEPS database.

Example 14-3 Example trace output

```
+43682387.0000        KBB_RAS1: ERROR (UNIT:ctdatabus INPUT,ERROR) (UNIT:ctsql
INPUT,ERROR)

(43682628.0013-1398:ctsqlaccesssql1.cpp,910,"CTSQLEvaluatorSQL1_i::AccessElemen
t::pullSequenceWithTimeout") HUB_REDBEARD(39): Rows returned: 1
(43682628.0014-138C:ctsqlstatement.cpp,199,"SQLStatement::SQLStatement")
TEPS2(69): SELECT ID, AFFINITIES, AUTH, AUTHEX, NAME, TEXT, LSTUSRPRF, LSTDATE
FROM KFWUSER WHERE (ID = 'SYSADMIN')
(43682628.0015-B7C:ctsqlaccessodbc.cpp,1007,"CTSQLEvaluatorODBC_i::AccessElemen
t::pullSequenceWithTimeout") TEPS2(69): Rows returned: 1
```

If the issue is with the **tacmd viewSit** command and the trace level is increased, then looking in the kuiras1.log will show output similar to Example 14-4.

Example 14-4 Example trace output

```
(43692109.0035-7B4:kuiviewsit.cpp,103,"viewsit") Compiled: Oct 25 2005 21:39:28
+43692109.0035      Level=1.2, Comp=*
(43692109.0036-7B4:kuiviewsit.cpp,103,"viewsit") Active RAS1 Classes: EVERYT
EVERYE EVERYU
(43692109.0037-7B4:kuiviewsit.cpp,103,"viewsit") Entry
(43692109.0038-7B4:kuiviewsit.cpp,185,"viewsit") SQL QUERY FOR SITDESC SELECT
ADVISE,AFFINITIES,ALERTLIST,AUTOSOPT,AUTOSTART,CMD,DESTNODE,HUB,LOCFLAG,LSTCCSI
D,LSTDATE,LSTRELEASE,LSTUSRPRF,NOTIFYARGS,NOTIFYOPTS,OBJECTLOCK,PDT,PRNAMES,QIB
SCOPE,REEV_DAYS,REEV_TIME,REFLEXOK,SENDMSGQ,SITINFO,SITNAME,SOURCE,TEXT FROM
O4SRV.TSITDESC WHERE SITNAME ='testSituation'
(43692109.0039-7B4:kuiviewsit.cpp,432,"viewsit") Exit: 0x0
(43692109.003A-7B4:kuitacmdmain.cpp,200,"main") Password exists
(43692109.003B-7B4:kuiviewsit.cpp,103,"viewsit") Entry
(43692109.003C-7B4:kuiviewsit.cpp,237,"viewsit") SQL QUERY FOR Distribution
SELECT OBJNAME,NODEL FROM O4SRV.TOBJACCL WHERE OBJNAME='testSituation' AND
SYSTEM.PARMA("QIBNODE", "QOMEGAVIEW", 32)
(43692109.003D-7B4:kuiviewsit.cpp,432,"viewsit") Exit: 0x0
(43692109.003E-7B4:kuitacmdmain.cpp,200,"main") Password exists
(43692109.003F-7B4:kuitacmdmain.cpp,211,"main") Want to send soap request
(43692109.0040-7B4:RAS1,400,"CTBLD")

Followed by the output in the log

(43692109.0055-7B4:kuiviewsit.cpp,649,"processResponse") Distribution Info
Primary:REDBEARD:NT
(43692109.0056-7B4:kuiviewsit.cpp,661,"processResponse") SITNAME Info
testSituation
(43692109.0057-7B4:kuiviewsit.cpp,685,"processResponse") AFFINITIES
(43692109.0058-7B4:kuiviewsit.cpp,696,"processResponse") PDT *IF *VALUE
NT_Services.Service_Name *EQ 'Alerter' *AND *VALUE NT_Services.Start_Type *EQ
'Automatic' *AND *VALUE NT_Services.Current_State *EQ 'Stopped'
```

```
(43692109.0059-7B4:kuiviewsit.cpp,709,"processResponse") SAMPLING INTERVAL 0
003000
(43692109.005A-7B4:kuiviewsit.cpp,722,"processResponse") AUTOSTART *YES
(43692109.005B-7B4:kuiviewsit.cpp,733,"processResponse") ADVISE *NONE
(43692109.005C-7B4:kuiviewsit.cpp,744,"processResponse") CMD *NONE
(43692109.005D-7B4:kuiviewsit.cpp,755,"processResponse") AUTOSOPT NNY
(43692109.005E-7B4:kuiviewsit.cpp,792,"processResponse") QIBSCOPE E
(43692109.005F-7B4:kuiviewsit.cpp,803,"processResponse") SENDMESGQ *NONE
(43692109.0060-7B4:kuiviewsit.cpp,835,"processResponse") LSTCCSID en_US
(43692109.0061-7B4:kuiviewsit.cpp,846,"processResponse") LSTCCSID
1051102100329000
(43692109.0062-7B4:kuiviewsit.cpp,857,"processResponse") LSTRELEASE V100
(43692109.0063-7B4:kuiviewsit.cpp,868,"processResponse") LSTUSRPRF SYSADMIN
(43692109.0064-7B4:kuiviewsit.cpp,934,"processResponse") SITINFO ~
```

One thing to note is that when the trace level is increased the level of details in the log is greatly increased. With this in mind when using these logs, you will see some entries that appear to be errors that can be ignored, such as communication errors that are logged about an interface that is not configured for the component. This is why it is important to know the issue being addressed and to have a knowledge of the system to know what can be ignored.

Additionally, it is important to understand the configuration of the servers when troubleshooting a server issue. Some things to consider:

- Security: enabled/disabled?
- Protocol in use?
- Firewall in use?
- Is it configured to use IP or host name?
- What port number is used?

14.4 Using the product documentation

The product documentation can be quite helpful when troubleshooting problems. *IBM Tivoli Monitoring Problem Determination Guide*. GC32-9458, covers many topics related to troubleshooting the IBM Tivoli Monitoring 6.1 components. This guide also contains details about using the product log files, messaging, and some environment variables that can control the product.

Additionally, many of the agent User's Guides include problem determination appendices, which have information specific to the agent about which the guide is written.

14.5 Tools

IBM Tivoli Monitoring 6.1 comes with several tools that can be used to troubleshoot problems and collect data to send to IBM Customer Support. These tools are helpful in reading logs and collecting data about the IBM Tivoli Monitoring environment.

- **digup**

  Used to collect information about the IBM Tivoli Monitoring 6.1 environment on the TEMS where it is run. This powerful command creates an image of the TEMS install. The options that can be used with the **digup** command are:

  ```
  digup [-h candle_directory] [-a] [-s] [-k] [-i] [-t] [-l label]
  ```

-a	Builds a tar file of all your site information.
-s	Gets a one-line OS Summary
-k	Keeps the working directory after completion
-i	Ignores agent logs when building the site info tar file
-t	Gets cms tables
-l	Defines a label or text string identifier to display on the output
-h	Specifies a CandleHome directory

- Log reader

  The log reader, which is accessible from the Manage Tivoli Enterprise Monitoring Services, is used to open the IBM Tivoli Monitoring 6.1 logs and can convert the hexadecimal timestamp into a human-readable format.

Other commands, such as **cinfo**, can be quite helpful when looking at running processes on the TEMS/TEPS.

14.6 Sample problem scenarios

This section provides some samples of issues encountered in the product's development and testing, as well as the IBM Tivoli Monitoring 6.1 Beta program. We include the following samples:

- Installation issues
  - Installation of the TEPS on Linux
  - Installation of the TEC event synchronization
- Logon to the Tivoli Enterprise Portal client failures
- Events being sent with the "wrong" severity
- Command line fails to list situations (tacmd)
- TEP desktop shows an agent incorrectly but the TEP browser works fine

14.6.1 Installation issues

Installation problems, the first issues that will be encountered, must be resolved quickly so that the product can be installed and operational. Two issues that were encountered in the IBM Tivoli Monitoring 6.1 Beta program were the installation of the TEPS on Linux and the installation of the TEC event synchronization.

Installing the TEPS on Linux

With this release of IBM Tivoli Monitoring 6.1, the TEPS is supported on Linux (SLES 9 and RHES 4). Several factors can cause issues when installing on Linux, and some will be picked up by the prerequisite check such as the requirement for Korn shell (ksh). In this case, the install has passed the prerequisite checks and appears to install; however, a problem is encountered with the database portion of the install.

What should be checked?

- Logs
  - /opt/IBM/ITM/logs/db2prep.log
  - /opt/IBM/ITM/logs/InstallPresentation.log
- Ensure that DB2 is installed and the instance created and running.
  - Log on to the Linux system and switch users to the DB2 instance owner, as shown in Example 14-5.

Example 14-5 Checking DB2 installation and instance existence

```
#su - db2inst1
db2inst1>db2ilist
db2inst1
db2inst1>db2start
SQL1026N  The database manager is already active
db2inst1>db2 list db directory
Does it show the TEPS database (assuming post install)?
System Database Directory

 Number of entries in the directory = 1

Database 1 entry:

 Database alias                       = TEPS
 Database name                        = TEPS
 Local database directory             = /home/db2inst1
```

Assuming that this fails to show the TEPS database, proceed to look at the db2prep.log and see why the database was not created. The db2prep.log shows that the DB2 instance was not running at the time of the install.

Example 14-6 /opt/IBM/ITM/logs/db2prep.log

```
KFWDB2PREP done
Attaching to DB2 instance db2inst1...
SQL1032N  No start database manager command was issued.  SQLSTATE=57019
Failed to attach to the db2 instance
Please check the followings:
Check the existence of the db2inst1 instance.
Check the instance is started and running.
Check the Admin User  db2inst1 is defined.
Check the Admin User password is correct.
```

The solution is quite simple: Start the DB2 instance and reconfigure the TEPS.

Example 14-7 Solution: reconfigure TEPS

```
#su - db2inst1
db2inst1>db2start
SQL1063N  DB2START processing was successful.
db2inst1>exit
#cd /opt/IBM/ITM/bin
#./itmcmd config -A cq

Enter requested informaiton and the d/opt/IBM/ITM/logs/db2prep.log should show:

Attaching to DB2 instance db2inst1...

   Instance Attachment Information

 Instance server        = DB2/LINUX 8.2.0
 Authorization ID       = DB2INST1
 Local instance alias   = DB2INST1

Creating the database TEPS...
DB20000I The CREATE DATABASE command completed successfully.
Connecting to the database as db2inst1...

   Database Connection Information

 Database server        = DB2/LINUX 8.2.0
 SQL authorization ID   = DB2INST1
 Local database alias   = TEPS

Granting rights to itmuser...
DB20000I  The SQL command completed successfully.
```

```
Resetting connection...
DB20000I  The SQL command completed successfully.
Testing Connection to TEPS with itmuser UserID...

   Database Connection Information

 Database server        = DB2/LINUX 8.2.0
 SQL authorization ID   = ITMUSER
 Local database alias   = TEPS

KFWDB2PREP done
```

Installing the TEC event synchronization

Before this component can be installed on the IBM Tivoli Enterprise Console Server system, IBM Tivoli Enterprise Console 3.9 with 3.9-TEC-FP03 must be installed. The issue here is that the TEC event synchronization installation is failing due to missing prerequisites. The installation returns this message:

```
The installation is unable to continue because a TEC server with 3.9.0
installation with Fix Pack3 or higher cannot be found.
```

The IBM Tivoli Enterprise Console system shows that IBM Tivoli Enterprise Console 3.9 and 3.9-TEC-FP03 are installed, so how do you proceed?

- Logs located.
  - Windows: %TEMP%/itm_tec_event_sync_install.log
  - UNIX: /tmp/itm_tec_event_sync_install.log
- Check to make sure that the prerequisites are installed.
  - Log on to the Tivoli Management Server (TMR) where the IBM Tivoli Enterprise Console is located and run the following commands shown in Example 14-8 (assuming UNIX TMR) and ensure that IBM Tivoli Enterprise Console 3.9 and 3.9-TEC-FP03 are installed.

Example 14-8 Check to make sure that the prerequisites are installed

```
. /etc/Tivoli/setup_env.sh
wlsinst -a | grep "Enterprise Console Server 3.9"
IBM Tivoli Enterprise Console Server 3.9 Upgrade
    suntmr01 solaris2

3.9.0 Tivoli Enterprise Console Server Fix Pack 1
    suntmr01 solaris2

3.9.0 Tivoli Enterprise Console Server Fix Pack 3
    suntmr01 solaris2
```

Now, see what the TEC event synchronization logs show. The sample log file in Example 14-9 comes from a Solaris TMR server.

Example 14-9 TEC event synchronization logs

```
(Oct 18, 2005 12:33:26 PM), null, com.ibm.opms.services.TMEService, dbg, OS
type: SunOS
(Oct 18, 2005 12:33:26 PM), null, com.ibm.opms.services.TMEService, dbg, Setup
environment filename: /etc/Tivoli/setup_env.sh
(Oct 18, 2005 12:33:26 PM), null, com.ibm.opms.wizard.actions.GetTmeInfo, dbg,
GetTmeInfo - setupEnvLocn is: /etc/Tivoli/setup_env.sh
(Oct 18, 2005 12:33:26 PM), null, com.ibm.opms.wizard.actions.GetTmeInfo, dbg,
/etc/Tivoli/setup_env.sh exists
(Oct 18, 2005 12:33:26 PM), null, com.ibm.opms.services.TMEService, dbg,
TMEService: entered
......
(Oct 18, 2005 12:33:26 PM), null, com.ibm.opms.wizard.actions.GetTmeInfo, dbg,
GetTmeInfo - tmeBinDir is: /usr/local/Tivoli/bin/solaris2
(Oct 18, 2005 12:33:26 PM), null, com.ibm.opms.wizard.actions.GetTmeInfo, dbg,
GetTmeInfo - tecDir is: /usr/local/Tivoli/bin/solaris2/TME/TEC
(Oct 18, 2005 12:33:26 PM), null, com.ibm.opms.wizard.actions.GetTmeInfo, dbg,
GetTmeInfo - Looking for /usr/local/Tivoli/bin/solaris2/TME/TEC/TECX030900.sys
(Oct 18, 2005 12:33:26 PM), null, com.ibm.opms.wizard.actions.GetTmeInfo, dbg,
GetTmeInfo - Looking for /usr/local/Tivoli/bin/solaris2/TME/TEC/TECS030900.sys
(Oct 18, 2005 12:33:26 PM), null, com.ibm.opms.wizard.actions.GetTmeInfo, dbg,
GetTmeInfo - Found /usr/local/Tivoli/bin/solaris2/TME/TEC/TECS030900.sys
(Oct 18, 2005 12:33:26 PM), null, com.ibm.opms.wizard.actions.GetTmeInfo, dbg,
GetTmeInfo - /usr/local/Tivoli/bin/solaris2/TME/TEC/TECS030900.sys size = 100
(Oct 18, 2005 12:33:26 PM), null, com.ibm.opms.wizard.actions.GetTmeInfo, err,
GetTmeInfo - /usr/local/Tivoli/bin/solaris2/TME/TEC/TECS030900.sys size of 100
indicates Fix Pack 3 or higher is NOT installed
(Oct 18, 2005 12:33:26 PM), null, com.ibm.opms.wizard.actions.GetTmeInfo, err,
GetTmeInfo: rc: 951
```

The installation is checking to make sure that TECS030900.sys is 130 bytes or greater, as this indicates that 3.9-TEC-FP03 is installed. If this file is smaller than 130 bytes and 3.9-TEC-FP03 is installed, then something has changed the size of this file. In this case, an interim fix has modified this file incorrectly and the file can be modified back because it contains 13 lines of 10 bytes each. (The text in the file is `123456789`).

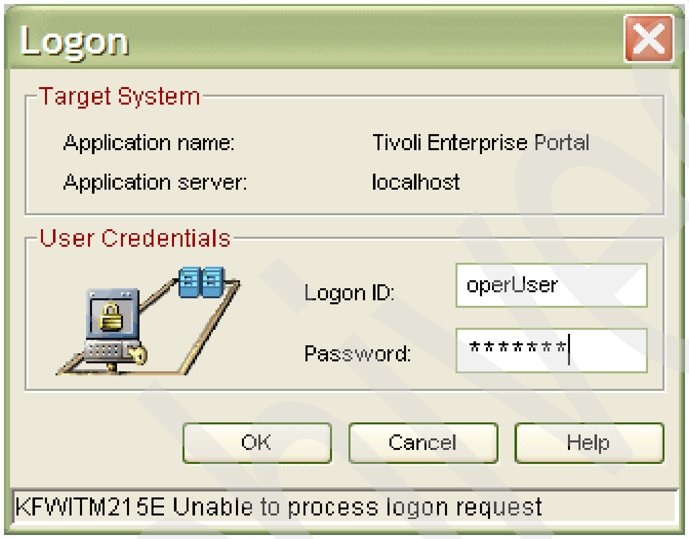
The solution is to add three lines of `123456789` to the file, which corrected the problem so the installation was able to proceed.

Note: Confirmation is required that the prerequisites are installed before modifying this file. You should contact IBM Support for assistance if you have any questions.

14.6.2 Logging on to the Tivoli Enterprise Portal client fails

A user attempts to log on to the Tivoli Enterprise Portal via a browser and is denied a logon. What is wrong?

The user starts the browser interface and points to the Tivoli Enterprise Portal Server. When the user ID and password are entered, the result is a logon failure as seen in Figure 14-2.



Application name: Tivoli Enterprise Portal
Application server: localhost
Logon ID: operUser
Password: *******

KFWITM215E Unable to process logon request

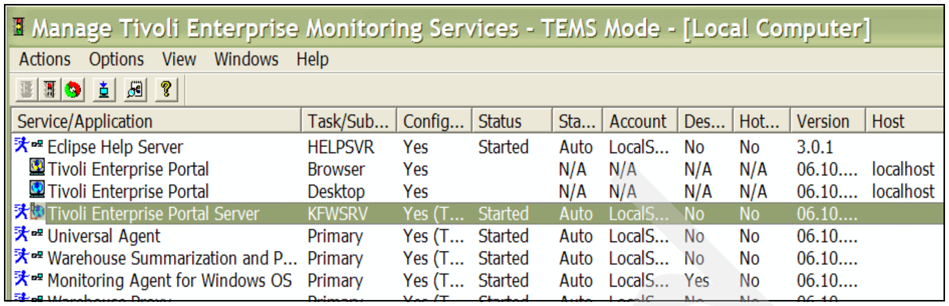
Figure 14-2 Logon failure message

This issue can isolated and corrected by using these steps to troubleshoot it:

- Is the Tivoli Enterprise Portal Server running?
- Is the TEPS database up?
- Is the user defined to the TEPS?
- Is the password entered correctly?
- Is security enabled on the TEMS?
- Is the TEPS connecting to the TEMS?

The location of the TEPS determines the method used to tell whether the server is running:

- Windows
   a. Open the Manage Tivoli Enterprise Monitoring Services menu and check to see whether the TEPS is started, as seen in Figure 14-3 on page 741.



Manage Tivoli Enterprise Monitoring Services - TEMS Mode - [Local Computer]

Service/Application	Task/Sub...	Config...	Status	Sta...	Account	Des...	Hot...	Version	Host
Eclipse Help Server	HELPSVR	Yes	Started	Auto	LocalS...	No	No	3.0.1	
Tivoli Enterprise Portal	Browser	Yes		N/A	N/A	N/A	N/A	06.10....	localhost
Tivoli Enterprise Portal	Desktop	Yes		N/A	N/A	N/A	N/A	06.10....	localhost
Tivoli Enterprise Portal Server	KFWSRV	Yes (T...	Started	Auto	LocalS...	No	No	06.10....	
Universal Agent	Primary	Yes (T...	Started	Auto	LocalS...	No	No	06.10....	
Warehouse Summarization and P...	Primary	Yes (T...	Started	Auto	LocalS...	No	No	06.10....	
Monitoring Agent for Windows OS	Primary	Yes (T...	Started	Auto	LocalS...	Yes	No	06.10....	

Figure 14-3 Manage Tivoli Monitoring Services menu

b. From a command window on the TEPS system, type:

```
c:\IBM\ITM\InstallITM\kincinfo -r
```

Example 14-10 shows the output.

Example 14-10 kincinfo -r output

```
******Fri Nov 04 10:45:15 Eastern Standard Time 2005 *****************
User      : Administrator Group    : NA
Host Name : REDBEARD     Installer: Ver: NOVALUE
CandleHome: C:\IBM\ITM
*********************************************************************
Host Prod  PID     Owner                  Start     Status         Task
REDBEARD FW 756 NT AUTHORITY\SYSTEM    9:21:19  ..Running      KFWSRV

This is the TEPS process on Windows.
```

- Linux

  a. Use the command **cinfo** to determine whether the TEPS is running. Log on to the TEPS server and run:

```
/opt/IBM/ITM/bin/cinfo -r
```

Example 14-11 shows the output.

Example 14-11 cinfo -r output

```
*********** Fri Nov  4 10:54:42 EST 2005 *****************
User      : root         Group: root pkcs11
Host name : fins         Installer Lvl: 400 / 100
CandleHome: /opt/IBM/ITM
***********************************************************
Host      Prod     PID     Owner    Start    ID  ..Status
fins      cq       31099   root     10:56 None ..running

This is the TEPS process on Linux - cq.
```

The TEPS logs shows whether the database is down (Example 14-12).

Example 14-12 TEPS log with DB2 down

```
(Friday, November 04, 2005,
10:54:28-{A48}ctsqlconnectionodbc.cpp,116,"CTSQLEvaluatorODBC_i::Connection::sq
lErrorCheck") SQLDriverConnect rc=-1: SQL_ERROR
+436B8434.0000 SQLSTATE: 08001, ERR: -1032, MSG: [IBM][CLI Driver] SQL1032N  No
start database manager command was issued.  SQLSTATE=57019
(Friday, November 04, 2005,
10:54:28-{A48}ctpropertysequence.cpp,560,"CTPropertySequence::Dump")   --->
name = EXCEPTION: SQL Exception: accessElement = 0370EC80, connection =
03714FD8
(Friday, November 04, 2005,
10:54:28-{A48}ctpropertysequence.cpp,560,"CTPropertySequence::Dump")   ----->
-2101 = -1 L
(Friday, November 04, 2005,
10:54:28-{A48}ctpropertysequence.cpp,560,"CTPropertySequence::Dump")   ----->
-2102 = "SQLDriverConnect rc=-1: SQL_ERROR
+436B8434.0003 SQLSTATE: 08001, ERR: -1032, MSG: [IBM][CLI Driver] SQL1032N  No
start database manager command was issued.  SQLSTATE=57019
+436B8434.0003 "
(Friday, November 04, 2005,
10:54:28-{177C}ctpropertysequence.cpp,560,"CTPropertySequence::Dump")   --->
name = EXCEPTION: Previous SQL Exception: connection = 03714FD8
(Friday, November 04, 2005,
10:54:28-{177C}ctpropertysequence.cpp,560,"CTPropertySequence::Dump")   ----->
-2101 = -1 L
(Friday, November 04, 2005,
10:54:28-{177C}ctpropertysequence.cpp,560,"CTPropertySequence::Dump")   ----->
-2102 = "SQLDriverConnect rc=-1: SQL_ERROR
+436B8434.0006 SQLSTATE: 08001, ERR: -1032, MSG: [IBM][CLI Driver] SQL1032N  No
start database manager command was issued.  SQLSTATE=57019
+436B8434.0006 "
```

When logging on to the TEPS, the first check is made to the TEPS table that defines the user to the system: TEPS.KFWUSER. This check can be seen in the TEPS logs as seen in Example 14-13.

Example 14-13 Two examples from TEPS log on user

```
User not defined:

(43503C88.0000-1228:ctauthorizationevaluator_i.cpp,727,"CTAuthorization::Evalua
tor_i::executeQuery") Invalid Userid <operUser>
(43503C88.0001-1228:ctdatabusmanager_i.cpp,736,"CTDataBus_i::Manager_i::Data::e
xecuteRequest") EXCEPTION: ::CTProperty::PropertyBasedException -
executeRequest
```

```
User defined - trace level increased:

(43682628.0013-1398:ctsqlaccesssql1.cpp,910,"CTSQLEvaluatorSQL1_i::AccessElemen
t::pullSequenceWithTimeout") HUB_REDBEARD(39): Rows returned: 1
(43682628.0014-138C:ctsqlstatement.cpp,199,"SQLStatement::SQLStatement")
TEPS2(69): SELECT ID, AFFINITIES, AUTH, AUTHEX, NAME, TEXT, LSTUSRPRF, LSTDATE
FROM KFWUSER WHERE (ID = 'operUser')
(43682628.0015-B7C:ctsqlaccessodbc.cpp,1007,"CTSQLEvaluatorODBC_i::AccessElemen
t::pullSequenceWithTimeout") TEPS2(69): Rows returned: 1
```

The password will not be seen in the logs. This will be the password of the user on the TEMS system. If the user exists and the password is questioned, the user should try again with the known password or attempt resetting the password on the TEMS operating system. The password is not stored in the TEPS or TEMS, and the user is validated on the TEMS at logon with the operating system.

You can check whether security is enabled on the TEMS system in the logs and by looking at the TEMS:

- Entry in the TEMS log:

  ```
  (Tuesday, November 01, 2005, 21:10:32-{D00}kbbssge.c,52,"BSS1_GetEnv")
  CMS_VALIDATE="YES"
  ```

- Viewing the TEMS settings on UNIX using itmcmd:

  ```
  /opt/IBM/ITM/bin/itmcmd config -S -g -t fins | grep SECURITY
  SECURITY=NO
  ```

If the TEPS and TEMS are not communicating, the error message should be different (something like `KFWITM001W Unable to connect to Tivoli Enterprise Portal Server`). If the TEPS shows that it is up, then a likely cause is that the TEMS is down or not responding to the TEPS logon request.

The solution to this situation depends on the results of the troubleshooting:

- Start the TEPS.
- Start the database.
- Add the user or use an existing user.
- Use the correct password.
- If security is not enabled, do not use a password.
- Restart the TEMS.

14.6.3 Events being sent with the "wrong" severity

Events being generated by the IBM Tivoli Monitoring 6.1 are all coming in to the Tivoli Enterprise Console Server as severity UNKNOWN. The IBM Tivoli Enterprise Console is configured with the IBM Tivoli Monitoring 6.1 baroc files, and events are flowing to the IBM Tivoli Enterprise Console. The situations generating the events are not UNKNOWN severity, so what is happening?

Event severity is generated in three ways for IBM Tivoli Enterprise Console events in IBM Tivoli Monitoring 6.1:

- The situation has a severity assigned. SITINFO column defines:
  - ITM 6.1 Informational maps to HARMLESS
  - ITM 6.1 Warning maps to WARNING
  - ITM 6.1 Critical maps to CRITICAL
- The situation suffix controls the mapping:
  - Name ending in Critical, Crit, or Cri maps to CRITICAL
  - Name ending in Warning or Warn maps to WARNING
  - Other names map to UNKNOWN
- Situational mapping defined in tecserver.txt file on the TEMS. For example:

  ```
  testSituation=tecserver2,SEVERITY=CRITICAL
  ```

A quick method to test this is to modify the severity being sent by IBM Tivoli Monitoring 6.1 on the TEMS is to change the om_tec.config file and have the events sent to a file on the TEMS (Example 14-14).

Example 14-14 Test om_tec.config

```
TestMode=YES
ServerLocation=c:\IBM\ITM\cms\TECLIB\tec_event.out
#ServerLocation=tec_serverhost
```

This generates events to the file defined in ServerLocation, and this file can be viewed to check the severity before it is sent to IBM Tivoli Enterprise Console.

Example 14-15 Sample events from the local test file

```
ITM_NT_Process;source='ITM';sub_source='Primary:REDBEARD:NT';cms_hostname='redb
eard.raleigh.ibm.com';cms_port='3661';integration_type='N';master_reset_flag=''
;appl_label='';situation_name='NT_Missing_Process_Warning';situation_origin='Pr
imary:REDBEARD:NT';situation_time='11/04/2005
09:22:06.006';situation_status='Y';hostname='REDBEARD';origin='9.27.203.252';ad
apter_host='HUB_REDBEARD';severity='WARNING';date='11/04/2005';msg='NT_Missing_
Process_Warning';situation_displayitem='';process_name='Alerter';server_name='P
rimary:REDBEARD:NT';situation_eventdata='~';END
ITM_NT_Services;source='ITM';sub_source='Primary:REDBEARD:NT';cms_hostname='red
beard.raleigh.ibm.com';cms_port='3661';integration_type='N';master_reset_flag='
';appl_label='';situation_name='testSituation';situation_origin='Primary:REDBEA
RD:NT';situation_time='11/04/2005
```

```
09:22:06.007';situation_status='Y';hostname='REDBEARD';origin='9.27.203.252';ad
apter_host='HUB_REDBEARD';severity='CRITICAL';date='11/04/2005';msg='testSituat
ion[(Service_Name="Alerter" AND Start_Type="Manual" AND Current_State="Stopped"
) ON Primary:REDBEARD:NT (Service_Name=Alerter Start_Type=Manual
Current_State=Stopped )]';situation_displayitem='';account_id='NT
AUTHORITY\LocalService';account_id_u='NT
AUTHORITY\LocalService';binary_path='C:\WINDOWS\System32\svchost.exe -k
LocalService';binary_path_u='C:\WINDOWS\System32\svchost.exe -k
LocalService';current_state='Stopped';display_name='Alerter';display_name_u='Al
erter';load_order_group='
';server_name='Primary:REDBEARD:NT';service_name='Alerter';start_type='Manual';
start_type_enum='Manual';timestamp='1051104092206000';situation_eventdata='~';E
ND
```

The example above shows two events: One is sent with the correct severity based on the situation name; it ends in Warning, so the severity is WARNING. The other situation is defined in the tecserver.txt file to be sent as CRITICAL, so the event is defined with a severity of CRITICAL. The situations may have a different severity when viewed on the TEPS navigator; however, the definition for the events is correct. In most cases, the severity in the situation should match the event severity for clarity when being viewed by an operator.

The solution is to test the events on the TEMS to see how they are generated. This will have the same severity on the IBM Tivoli Enterprise Console unless rules are in place on the IBM Tivoli Enterprise Console to change the severity.

14.6.4 Command line fails to list situations (tacmd)

When using the tacmd to view a situation, unexpected results are produced. The tacmd logs to install_dir\bin\kuiras1.log on Windows and, on UNIX, install_dir/logs/kuiras1.log. With increased trace settings, the situation can be seen in the logs; if there is an issue, it will not be seen in the logs.

In this log the trace is set to KBB_RAS1=ERROR(UNIT:KUI ALL) and the details of the situation can be seen as shown in Example 14-16.

Example 14-16 Situation seen in kuiras1.log

```
(43692109.0037-7B4:kuiviewsit.cpp,103,"viewsit") Entry
(43692109.0038-7B4:kuiviewsit.cpp,185,"viewsit") SQL QUERY FOR SITDESC SELECT
ADVISE,AFFINITIES,ALERTLIST,AUTOSOPT,AUTOSTART,CMD,DESTNODE,HUB,LOCFLAG,LSTCCSI
D,LSTDATE,LSTRELEASE,LSTUSRPRF,NOTIFYARGS,NOTIFYOPTS,OBJECTLOCK,PDT,PRNAMES,QIB
SCOPE,REEV_DAYS,REEV_TIME,REFLEXOK,SENDMSGQ,SITINFO,SITNAME,SOURCE,TEXT FROM
O4SRV.TSITDESC WHERE SITNAME ='testSituation'
(43692109.0039-7B4:kuiviewsit.cpp,432,"viewsit") Exit: 0x0
...
(43692109.0052-7B4:kuiviewsit.cpp,448,"processResponse") Entry
```

```
(43692109.0053-7B4:kuiviewsit.cpp,570,"processResponse") Number of Rows in
TOBJACCL for sitname  is 1
(43692109.0054-7B4:kuiviewsit.cpp,572,"processResponse") Number of Rows in
TSITDESC for sitname  is 1
(43692109.0055-7B4:kuiviewsit.cpp,649,"processResponse") Distribution Info
Primary:REDBEARD:NT
(43692109.0056-7B4:kuiviewsit.cpp,661,"processResponse") SITNAME Info
testSituation
(43692109.0057-7B4:kuiviewsit.cpp,685,"processResponse") AFFINITIES
(43692109.0058-7B4:kuiviewsit.cpp,696,"processResponse") PDT *IF *VALUE
NT_Services.Service_Name *EQ 'Alerter' *AND *VALUE NT_Services.Start_Type *EQ
'Automatic' *AND *VALUE NT_Services.Current_State *EQ 'Stopped'
(43692109.0059-7B4:kuiviewsit.cpp,709,"processResponse") SAMPLING INTERVAL 0
003000
(43692109.005A-7B4:kuiviewsit.cpp,722,"processResponse") AUTOSTART *YES
(43692109.005B-7B4:kuiviewsit.cpp,733,"processResponse") ADVISE *NONE
(43692109.005C-7B4:kuiviewsit.cpp,744,"processResponse") CMD *NONE
(43692109.005D-7B4:kuiviewsit.cpp,755,"processResponse") AUTOSOPT NNY
(43692109.005E-7B4:kuiviewsit.cpp,792,"processResponse") QIBSCOPE E
(43692109.005F-7B4:kuiviewsit.cpp,803,"processResponse") SENDMESGQ *NONE
(43692109.0060-7B4:kuiviewsit.cpp,835,"processResponse") LSTCCSID en_US
(43692109.0061-7B4:kuiviewsit.cpp,846,"processResponse") LSTCCSID
1051102100329000
(43692109.0062-7B4:kuiviewsit.cpp,857,"processResponse") LSTRELEASE V100
(43692109.0063-7B4:kuiviewsit.cpp,868,"processResponse") LSTUSRPRF SYSADMIN
(43692109.0064-7B4:kuiviewsit.cpp,934,"processResponse") SITINFO ~
```

If **`tacmd viewSit`** -s *situation_name* does not show a situation, then check to make sure that the situation exists using **`tacmd listSit`** and that the situation is defined on the Hub TEMS. If a situation is defined on a remote using **`tacmd createSit`**, it will not be viewable from the hub. The situation should be defined on the hub and distributed to a managed system on the remote TEMS.

14.6.5 TEP desktop shows an agent incorrectly

The TEP desktop client shows an agent with incorrect attribute group label names as seen in Figure 14-4. The TEP browser client shows the labels correctly, so what is wrong and how can this be corrected?

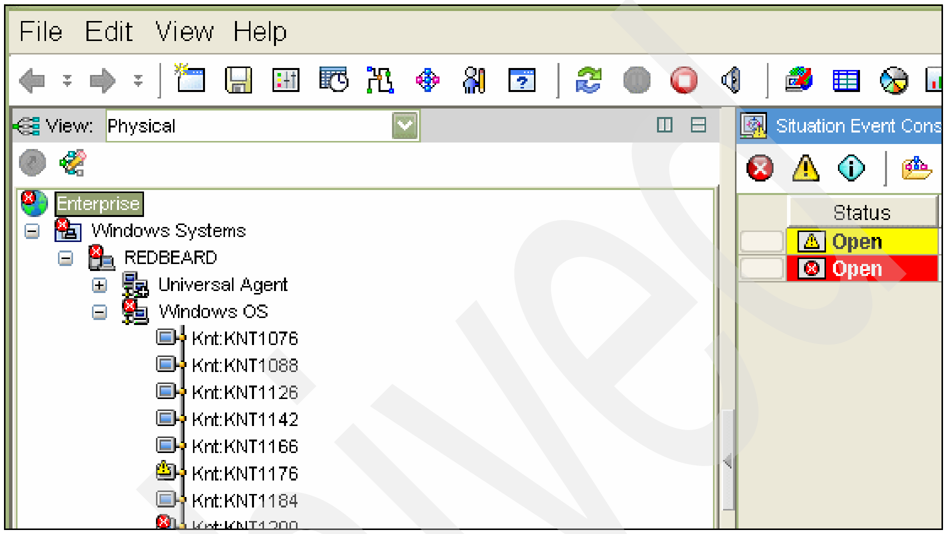


Figure 14-4 TEP Desktop client, incorrect Windows Agent attribute group labels

This can point to one of two issues:

- Application seeding is not complete.
- There is an incorrect classpath for the TEP desktop.

In this case, a review of the CMSseed.log shows Example 14-17.

Example 14-17 CMSseed.log

```
Addition of application support for component: knt
completed with rc: 0
C:\IBM\ITM\CNPS\sqllib\knt.sql
Output from the operation was written to log file:
C:\IBM\ITM\CNPS\logs\seedknt.log
```

Therefore, the Windows Monitoring Agent support appears to be loaded correctly in the TEMS, and these attribute group labels should show up correctly. In addition, the TEP browser shows the correct labels pointing to a problem with the desktop client.

Investigation of the desktop client log reveals the cause of the problem, as shown in Example 14-18 on page 748.

Example 14-18 TEP desktop client log - kcjras1.log

```
java.class.path =
cnp.jar;cnp_vbjorball.jar;ae.jar;kjrall.jar;cnp_jviewsall.jar;browser.jar;chart
.jar;terminal.jar;util.jar;icu4jm32.jar;deploy.jar;ka4_resources.jar;kit.jar;te
c_tap.jar;console.jar;ibmjsse.jar;jhall.jar;jsafe.zip;log4j-1.1.1.jar;launch.ja
r;nways_dep.jar;nvnways.jar;jcf.jar;jrim.jar;uif.jar;klz_resources.jar;kul_reso
urces.jar;kum_resources.jar;kux_resources.jar;.
......
(436ac67b.030a32c0-()Thread-3:CNPClientMgr,0,"CNPClientMgr.initializePackages()
") Package initialized: knt, version: 5.2.9.4
```

The java.class.path statement shows that there is not a knt_resources.jar file listed. This is the jar file used by the TEP desktop to load the Windows Agent views. This did not cause the knt package to fail initializing.

The solution is to modify the cnp.bat file, which is used by the TEP desktop client on Windows, and add the knt_resources.jar. It is also a good idea to verify that the knt_resources.jar exists on the TEP desktop client as well. Example 14-19 shows the updated cnp.bat classpath statement for the issue.

Example 14-19 Updated cnp.bat classpath statement

```
@set
CLASSPATH=cnp.jar;cnp_vbjorball.jar;ae.jar;kjrall.jar;cnp_jviewsall.jar;browser
.jar;chart.jar;terminal.jar;util.jar;
icu4jm32.jar;deploy.jar;ka4_resources.jar;kit.jar;tec_tap.jar;console.jar;ibmjs
se.jar;jhall.jar;jsafe.zip;log4j-1.1.1.jar
;launch.jar;nways_dep.jar;nvnways.jar;jcf.jar;jrim.jar;uif.jar;klz_resources.ja
r;knt_resources.jar;kul_resources.jar;kum_
resources.jar;kux_resources.jar
```

14.6.6 IBM Tivoli Monitoring 5.x Endpoint Agent issues

The IBM Tivoli Monitoring 5.x Endpoint Agent is used to send information from IBM Tivoli Monitoring 5.xx resource models to the IBM Tivoli Monitoring 6.1 environment. This enables the IBM Tivoli Monitoring 5.x environment to integrate into the IBM Tivoli Monitoring 6.1 environment. Several things are unique to the IBM Tivoli Monitoring 5.x Endpoint Agent that make deploying and troubleshooting it different from other components:

- Installs in the Tivoli Management Framework (TMF) environment.
- Deploys via the TMF mechanisms.
- Starts and stops with the IBM Tivoli Monitoring 5.x engine.
- Logs are located in the IBM Tivoli Monitoring 5.x structure.

This component installs using the TMF methods of **`winstall`** or the Tivoli Desktop. The installation logs to $DBDIR/KTM/logs/trace_ITM61Integration_Install____#_pid.log.($DBDIR is an environment variable of the TMF installation. This variable is set on all TMF ManagedNodes, including the TMR server.)

Example 14-20 Install log exiting with a good installation

```
+ '[' 0 -ne 0 ']'
+ echo 'Exiting installation after script'
+ exit 0
```

If the TEPS shows the IBM Tivoli Monitoring 5.x Endpoint Agent, but it does not appear correct, check to make sure that the IBM Tivoli Monitoring 5.x engine is running and that agents are running as well.

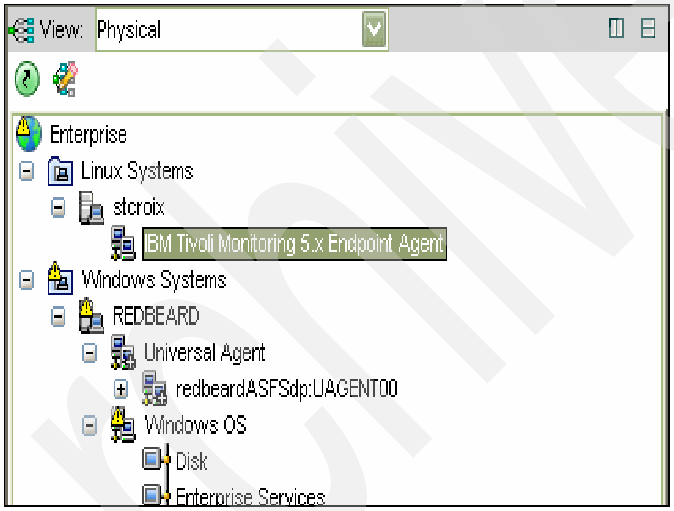


Figure 14-5 TEPS showing partial IBM Tivoli Monitoring 5.x Endpoint Agent

In the case of Figure 14-5 on page 749, the TEPS has to be refreshed by clicking on the green circle with the arrow in it. This result in Figure 14-6.

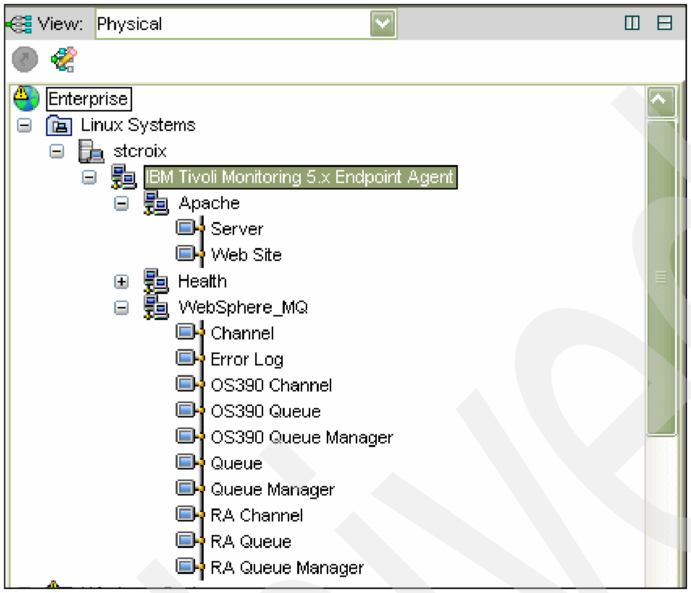


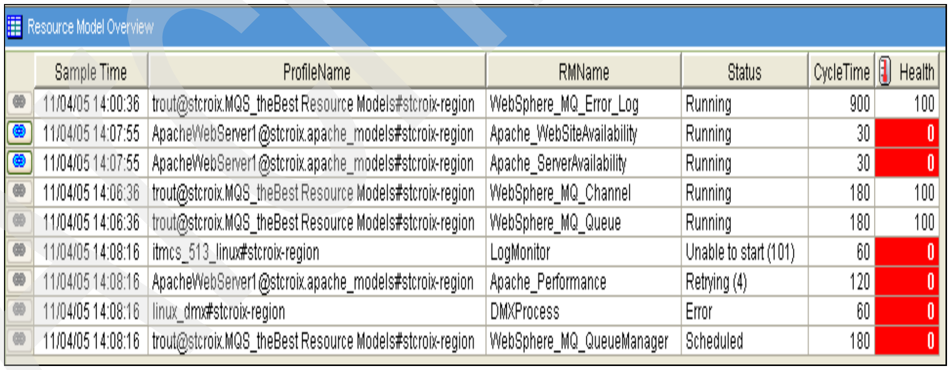
Figure 14-6   TEPS showing IBM Tivoli Monitoring 5.x Endpoint Agents

If the IBM Tivoli Monitoring 5.x Endpoint Agent is shown but some or all are grey, then several things can be checked, as shown in Example 14-21.

Example 14-21   Checking the running ITM 5.x engine and agents

```
From the TMR:
#wdmlseng -e stcroix
Forwarding the request to the endpoint:
stcroix  1904662587.7.522+#TMF_Endpoint::Endpoint#

The following profiles are running:

itmcs_513_linux#stcroix-region
        LogMonitor: Unable to start (101)
ApacheWebServer1@stcroix.apache_models#stcroix-region
        Apache_ServerAvailability: Running
        Apache_Performance: Retrying (4)
        Apache_WebSiteAvailability: Running
linux_dmx#stcroix-region
        DMXProcess: Error
trout@stcroix.MQS_theBest Resource Models#stcroix-region
        WebSphere_MQ_Queue: Running
        WebSphere_MQ_QueueManager: Scheduled
        WebSphere_MQ_Channel: Running
        WebSphere_MQ_Error_Log: Running

To see if the agents are running do this on the endpoint:
#ps -ef | grep ktm
root      9558  9557  0 14:00 ?        00:00:01
/opt2/tivoli/bin/linux-ix86/mrt/../TME/KTM/ktmcma
root      9561  9558  0 14:00 ?        00:00:00
/opt2/tivoli/bin/linux-ix86/mrt/../TME/KTM/ktmcma
root      9562  9561  0 14:00 ?        00:00:00
/opt2/tivoli/bin/linux-ix86/mrt/../TME/KTM/ktmcma
root      9563  9561  0 14:00 ?        00:00:00
/opt2/tivoli/bin/linux-ix86/mrt/../TME/KTM/ktmcma
root      9564  9561  0 14:00 ?        00:00:00
/opt2/tivoli/bin/linux-ix86/mrt/../TME/KTM/ktmcma
root      9565  9561  0 14:00 ?        00:00:00
/opt2/tivoli/bin/linux-ix86/mrt/../TME/KTM/ktmcma
root      9566  9561  0 14:00 ?        00:00:00
/opt2/tivoli/bin/linux-ix86/mrt/../TME/KTM/ktmcma
root      9568  9561  0 14:00 ?        00:00:00
/opt2/tivoli/bin/linux-ix86/mrt/../TME/KTM/ktmcma
```

In the case above, two resource models are in error state (DMXProcess and LogMonitor); because of this error state there is not a corresponding view of this in the TEPS Navigator. The Health view shows the resource model in the tabular overview and shows that it is in error, as seen in Figure 14-7.


Resource Model Overview

	Sample Time	ProfileName	RMName	Status	CycleTime	Health
	11/04/05 14:00:36	trout@stcroix.MQS_theBest Resource Models#stcroix-region	WebSphere_MQ_Error_Log	Running	900	100
	11/04/05 14:07:55	ApacheWebServer1@stcroix.apache_models#stcroix-region	Apache_WebSiteAvailability	Running	30	0
	11/04/05 14:07:55	ApacheWebServer1@stcroix.apache_models#stcroix-region	Apache_ServerAvailability	Running	30	0
	11/04/05 14:06:36	trout@stcroix.MQS_theBest Resource Models#stcroix-region	WebSphere_MQ_Channel	Running	180	100
	11/04/05 14:06:36	trout@stcroix.MQS_theBest Resource Models#stcroix-region	WebSphere_MQ_Queue	Running	180	100
	11/04/05 14:08:16	itmcs_513_linux#stcroix-region	LogMonitor	Unable to start (101)	60	0
	11/04/05 14:08:16	ApacheWebServer1@stcroix.apache_models#stcroix-region	Apache_Performance	Retrying (4)	120	0
	11/04/05 14:08:16	linux_dmx#stcroix-region	DMXProcess	Error	60	0
	11/04/05 14:08:16	trout@stcroix.MQS_theBest Resource Models#stcroix-region	WebSphere_MQ_QueueManager	Scheduled	180	0

Figure 14-7 Resource Model overview

In some cases, the **witm61agt** command has be rerun so that changes to the ITM engine can be picked up by the agent. For example, if the $LCF_DATDIR/LCFNEW directory is deleted, then the **witm61agt** command will have to be rerun because the KTM directory structure will have been removed and the agents will not be on the system to be started.

A

TEMA Software Package Definition examples

This appendix provides three examples of the Software Package Definiton (SPD) files that were used in Chapter 11, “Agent deployment using IBM Tivoli Configuration Manager Software Distribution” on page 633:

- Windows OS Agent SPD
- Universal Agent SPD for AIX
- AIX OS Agent SPD

These SPDs can be used to install TEMAs for these platforms via IBM Tivoli Configuration Manager.

You can download these and other TEMA SPDs and related silent response files and scripts from the IBM Redbooks Web site. For instructions on how to download these, refer to Appendix B, “Additional material” on page 803.

The Windows OS Agent SPD

Example A-1 shows the Windows OS Agent SPD (KNTWICMA.SPD).

Example: A-1 The Windows OS Agent SPD (KNTWICMA.SPD)

```
"TIVOLI Software Package v4.2.2 - SPDF"

package
   name = KNTWICMA
   title = "Windows OS Monitoring Agent"
   version = 6.1.0
   web_view_mode = hidden
   undoable = o
   committable = o
   history_reset = n
   save_default_variables = n
   creation_time = "2005-11-15 16:33:50"
   last_modification_time = "2005-11-15 16:35:24"

   default_variables
      SecondaryTEMSHostname = somehost.somewhere.com
      PortNumber = 1918
      IPPipePortNumber = 1918
      TEMSHostname = $(hostname)
      silentfiles_dir = C:\cm_inst\ITM\ITM6_SPBs\images\SilentFiles\Windows
      NetworkProtocol = IP.PIPE
      BackupNetworkProtocol = IP.UDP
      CandleEncryptionKey = IBMTivoliMonitoringEncryptionKey
      SNALogMode = CANCTDCS
      UseSecondaryTEMS = NO
      IPSPipePortNumber = 3660
      CandleHome = C:\IBM\ITM
      SNATPName = SNASOCKETS
      SNANetName = primary_net_name
      SNALUName = primary_lu_name
      source_dir = C:\cm_inst\ITM\ITM6_SPBs\images\RTM\win
   end

   move_removing_host = y
   no_check_source_host = y
   lenient_distribution = n
   default_operation = install
   server_mode = all
   operation_mode = not_transactional
   post_notice = n
   before_as_uid = 0
   skip_non_zero = n
   after_as_uid = 0
```

```
no_chk_on_rm = y
versioning_type = swd
package_type = refresh
sharing_control = none
stop_on_failure = y
condition = "$(os_name)  LIKE 'Win*'"

generic_container
   caption = "Silent Files"
   stop_on_failure = y

   add_directory
      stop_on_failure = y
      add = y
      replace_if_existing = y
      replace_if_newer = n
      remove_if_modified = y
      location = $(silentfiles_dir)
      translate = n
      destination = $(CandleHome)\InstSpbs
      descend_dirs = n
      remove_empty_dirs = y
      is_shared = n
      remove_extraneous = n
      substitute_variables = y
      unix_owner = root
      unix_user_id = 0
      unix_group_id = 0
      create_dirs = y
      remote = n
      compute_crc = n
      verify_crc = n
      delta_compressible = d
      temporary = n
      is_signature = n
      compression_method = deflated
      rename_if_locked = n

      add_file
         replace_if_existing = y
         replace_if_newer = y
         remove_if_modified = y
         name = silent_kntcma_install.txt
         translate = n
         destination = silent_kntcma_install.txt
         remove_empty_dirs = y
         is_shared = n
         remove_extraneous = n
         substitute_variables = y
         unix_owner = root
         unix_user_id = 0
         unix_group_id = 0
         create_dirs = y
         remote = n
         compute_crc = n
         verify_crc = n
         delta_compressible = d
         temporary = n
         is_signature = n
         compression_method = deflated
         rename_if_locked = n
      end


      add_file
         replace_if_existing = y
         replace_if_newer = y
         remove_if_modified = y
         name = silent_kntcma_uninstall.txt
         translate = n
         destination = silent_kntcma_uninstall.txt
         remove_empty_dirs = y
         is_shared = n
         remove_extraneous = n
         substitute_variables = y
         unix_owner = root
         unix_user_id = 0
         unix_group_id = 0
         create_dirs = y
         remote = n
         compute_crc = n
         verify_crc = n
         delta_compressible = d
         temporary = n
         is_signature = n
         compression_method = deflated
         rename_if_locked = n
      end

   end


   add_directory
      stop_on_failure = y
      add = y
      replace_if_existing = y
      replace_if_newer = n
      remove_if_modified = y
```

```
location = $(source_dir)
name = WINDOWS
translate = n
destination = $(CandleHome)\InstSpbs\WINDOWS
descend_dirs = n
remove_empty_dirs = y
is_shared = n
remove_extraneous = n
substitute_variables = n
unix_owner = root
unix_user_id = 0
unix_group_id = 0
create_dirs = y
remote = n
compute_crc = n
verify_crc = n
delta_compressible = d
temporary = n
is_signature = n
compression_method = deflated
rename_if_locked = n

add_file
   replace_if_existing = y
   replace_if_newer = n
   remove_if_modified = y
   name = KNTWICMA.cab
   translate = n
   destination = KNTWICMA.cab
   remove_empty_dirs = y
   is_shared = n
   remove_extraneous = n
   substitute_variables = n
   unix_owner = root
   unix_user_id = 0
   unix_group_id = 0
   create_dirs = y
   remote = n
   compute_crc = n
   verify_crc = n
   delta_compressible = d
   temporary = n
   is_signature = n
   compression_method = deflated
   rename_if_locked = n
end


add_file
```

```
   replace_if_existing = y
   replace_if_newer = n
   remove_if_modified = y
   name = KGLWICMA.cab
   translate = n
   destination = KGLWICMA.cab
   remove_empty_dirs = y
   is_shared = n
   remove_extraneous = n
   substitute_variables = n
   unix_owner = root
   unix_user_id = 0
   unix_group_id = 0
   create_dirs = y
   remote = n
   compute_crc = n
   verify_crc = n
   delta_compressible = d
   temporary = n
   is_signature = n
   compression_method = deflated
   rename_if_locked = n
end


add_file
   replace_if_existing = y
   replace_if_newer = n
   remove_if_modified = y
   name = KUIWICLI.cab
   translate = n
   destination = KUIWICLI.cab
   remove_empty_dirs = y
   is_shared = n
   remove_extraneous = n
   substitute_variables = n
   unix_owner = root
   unix_user_id = 0
   unix_group_id = 0
   create_dirs = y
   remote = n
   compute_crc = n
   verify_crc = n
   delta_compressible = d
   temporary = n
   is_signature = n
   compression_method = deflated
   rename_if_locked = n
end
```

```
add_file
   replace_if_existing = y
   replace_if_newer = n
   remove_if_modified = y
   name = INSTALL.cab
   translate = n
   destination = INSTALL.cab
   remove_empty_dirs = y
   is_shared = n
   remove_extraneous = n
   substitute_variables = n
   unix_owner = root
   unix_user_id = 0
   unix_group_id = 0
   create_dirs = y
   remote = n
   compute_crc = n
   verify_crc = n
   delta_compressible = d
   temporary = n
   is_signature = n
   compression_method = deflated
   rename_if_locked = n
end


add_file
   replace_if_existing = y
   replace_if_newer = n
   remove_if_modified = y
   name = INSTALLA.cab
   translate = n
   destination = INSTALLA.cab
   remove_empty_dirs = y
   is_shared = n
   remove_extraneous = n
   substitute_variables = n
   unix_owner = root
   unix_user_id = 0
   unix_group_id = 0
   create_dirs = y
   remote = n
   compute_crc = n
   verify_crc = n
   delta_compressible = d
   temporary = n
   is_signature = n
   compression_method = deflated
   rename_if_locked = n
end


add_file
   replace_if_existing = y
   replace_if_newer = n
   remove_if_modified = y
   name = INSTALLX.cab
   translate = n
   destination = INSTALLX.cab
   remove_empty_dirs = y
   is_shared = n
   remove_extraneous = n
   substitute_variables = n
   unix_owner = root
   unix_user_id = 0
   unix_group_id = 0
   create_dirs = y
   remote = n
   compute_crc = n
   verify_crc = n
   delta_compressible = d
   temporary = n
   is_signature = n
   compression_method = deflated
   rename_if_locked = n
end


add_file
   replace_if_existing = y
   replace_if_newer = n
   remove_if_modified = y
   name = KAXWICMA.cab
   translate = n
   destination = KAXWICMA.cab
   remove_empty_dirs = y
   is_shared = n
   remove_extraneous = n
   substitute_variables = n
   unix_owner = root
   unix_user_id = 0
   unix_group_id = 0
   create_dirs = y
   remote = n
   compute_crc = n
   verify_crc = n
   delta_compressible = d
   temporary = n
   is_signature = n
   compression_method = deflated
   rename_if_locked = n
end


add_file
   replace_if_existing = y
   replace_if_newer = n
   remove_if_modified = y
   name = KNSWICMA.cab
   translate = n
   destination = KNSWICMA.cab
   remove_empty_dirs = y
   is_shared = n
   remove_extraneous = n
   substitute_variables = n
   unix_owner = root
   unix_user_id = 0
   unix_group_id = 0
   create_dirs = y
   remote = n
   compute_crc = n
   verify_crc = n
   delta_compressible = d
   temporary = n
   is_signature = n
   compression_method = deflated
   rename_if_locked = n
end


add_file
   replace_if_existing = y
   replace_if_newer = n
   remove_if_modified = y
   name = 0x0409.ini
   translate = n
   destination = 0x0409.ini
   remove_empty_dirs = y
   is_shared = n
   remove_extraneous = n
   substitute_variables = n
   unix_owner = root
   unix_user_id = 0
   unix_group_id = 0
   create_dirs = y
```

```
      remote = n
      compute_crc = n
      verify_crc = n
      delta_compressible = d
      temporary = n
      is_signature = n
      compression_method = deflated
      rename_if_locked = n
   end


   add_file
      replace_if_existing = y
      replace_if_newer = n
      remove_if_modified = y
      name = instmsiw.exe
      translate = n
      destination = instmsiw.exe
      remove_empty_dirs = y
      is_shared = n
      remove_extraneous = n
      substitute_variables = n
      unix_owner = root
      unix_user_id = 0
      unix_group_id = 0
      create_dirs = y
      remote = n
      compute_crc = n
      verify_crc = n
      delta_compressible = d
      temporary = n
      is_signature = n
      compression_method = deflated
      rename_if_locked = n
   end


   add_file
      replace_if_existing = y
      replace_if_newer = n
      remove_if_modified = y
      name = ISScript9.Msi
      translate = n
      destination = ISScript9.Msi
      remove_empty_dirs = y
      is_shared = n
      remove_extraneous = n
      substitute_variables = n
      unix_owner = root
      unix_user_id = 0
      unix_group_id = 0
      create_dirs = y
      remote = n
      compute_crc = n
      verify_crc = n
      delta_compressible = d
      temporary = n
      is_signature = n
      compression_method = deflated
      rename_if_locked = n
   end


   add_file
      replace_if_existing = y
      replace_if_newer = n
      remove_if_modified = y
      name = lic.ini
      translate = n
      destination = lic.ini
      remove_empty_dirs = y
      is_shared = n
      remove_extraneous = n
      substitute_variables = n
      unix_owner = root
      unix_user_id = 0
      unix_group_id = 0
      create_dirs = y
      remote = n
      compute_crc = n
      verify_crc = n
      delta_compressible = d
      temporary = n
      is_signature = n
      compression_method = deflated
      rename_if_locked = n
   end


   add_file
      replace_if_existing = y
      replace_if_newer = n
      remove_if_modified = y
      name = non_ibm_license
      translate = n
      destination = non_ibm_license
      remove_empty_dirs = y
      is_shared = n
```

```
    remove_extraneous = n
    substitute_variables = n
    unix_owner = root
    unix_user_id = 0
    unix_group_id = 0
    create_dirs = y
    remote = n
    compute_crc = n
    verify_crc = n
    delta_compressible = d
    temporary = n
    is_signature = n
    compression_method = deflated
    rename_if_locked = n
end


add_file
    replace_if_existing = y
    replace_if_newer = n
    remove_if_modified = y
    name = notices
    translate = n
    destination = notices
    remove_empty_dirs = y
    is_shared = n
    remove_extraneous = n
    substitute_variables = n
    unix_owner = root
    unix_user_id = 0
    unix_group_id = 0
    create_dirs = y
    remote = n
    compute_crc = n
    verify_crc = n
    delta_compressible = d
    temporary = n
    is_signature = n
    compression_method = deflated
    rename_if_locked = n
end


add_file
    replace_if_existing = y
    replace_if_newer = n
    remove_if_modified = y
    name = RenameList.txt
    translate = n
```

```
   destination = RenameList.txt
   remove_empty_dirs = y
   is_shared = n
   remove_extraneous = n
   substitute_variables = n
   unix_owner = root
   unix_user_id = 0
   unix_group_id = 0
   create_dirs = y
   remote = n
   compute_crc = n
   verify_crc = n
   delta_compressible = d
   temporary = n
   is_signature = n
   compression_method = deflated
   rename_if_locked = n
end


add_file
   replace_if_existing = y
   replace_if_newer = n
   remove_if_modified = y
   name = README.TXT
   translate = n
   destination = README.TXT
   remove_empty_dirs = y
   is_shared = n
   remove_extraneous = n
   substitute_variables = n
   unix_owner = root
   unix_user_id = 0
   unix_group_id = 0
   create_dirs = y
   remote = n
   compute_crc = n
   verify_crc = n
   delta_compressible = d
   temporary = n
   is_signature = n
   compression_method = deflated
   rename_if_locked = n
end


add_file
   replace_if_existing = y
   replace_if_newer = n
   remove_if_modified = y
   name = SETUP.BMP
   translate = n
   destination = SETUP.BMP
   remove_empty_dirs = y
   is_shared = n
   remove_extraneous = n
   substitute_variables = n
   unix_owner = root
   unix_user_id = 0
   unix_group_id = 0
   create_dirs = y
   remote = n
   compute_crc = n
   verify_crc = n
   delta_compressible = d
   temporary = n
   is_signature = n
   compression_method = deflated
   rename_if_locked = n
end


add_file
   replace_if_existing = y
   replace_if_newer = n
   remove_if_modified = y
   name = setup.exe
   translate = n
   destination = setup.exe
   remove_empty_dirs = y
   is_shared = n
   remove_extraneous = n
   substitute_variables = n
   unix_owner = root
   unix_user_id = 0
   unix_group_id = 0
   create_dirs = y
   remote = n
   compute_crc = n
   verify_crc = n
   delta_compressible = d
   temporary = n
   is_signature = n
   compression_method = deflated
   rename_if_locked = n
end
```

```
add_file
   replace_if_existing = y
   replace_if_newer = n
   remove_if_modified = y
   name = Setup.ini
   translate = n
   destination = Setup.ini
   remove_empty_dirs = y
   is_shared = n
   remove_extraneous = n
   substitute_variables = n
   unix_owner = root
   unix_user_id = 0
   unix_group_id = 0
   create_dirs = y
   remote = n
   compute_crc = n
   verify_crc = n
   delta_compressible = d
   temporary = n
   is_signature = n
   compression_method = deflated
   rename_if_locked = n
end


add_file
   replace_if_existing = y
   replace_if_newer = n
   remove_if_modified = y
   name = setup.iss
   translate = n
   destination = setup.iss
   remove_empty_dirs = y
   is_shared = n
   remove_extraneous = n
   substitute_variables = n
   unix_owner = root
   unix_user_id = 0
   unix_group_id = 0
   create_dirs = y
   remote = n
   compute_crc = n
   verify_crc = n
   delta_compressible = d
   temporary = n
   is_signature = n
   compression_method = deflated
   rename_if_locked = n
```

```
end


add_file
   replace_if_existing = y
   replace_if_newer = n
   remove_if_modified = y
   name = Setup.skin
   translate = n
   destination = Setup.skin
   remove_empty_dirs = y
   is_shared = n
   remove_extraneous = n
   substitute_variables = n
   unix_owner = root
   unix_user_id = 0
   unix_group_id = 0
   create_dirs = y
   remote = n
   compute_crc = n
   verify_crc = n
   delta_compressible = d
   temporary = n
   is_signature = n
   compression_method = deflated
   rename_if_locked = n
end


add_file
   replace_if_existing = y
   replace_if_newer = n
   remove_if_modified = y
   name = TMV610.msi
   translate = n
   destination = TMV610.msi
   remove_empty_dirs = y
   is_shared = n
   remove_extraneous = n
   substitute_variables = n
   unix_owner = root
   unix_user_id = 0
   unix_group_id = 0
   create_dirs = y
   remote = n
   compute_crc = n
   verify_crc = n
   delta_compressible = d
   temporary = n
```

```
         is_signature = n
         compression_method = deflated
         rename_if_locked = n
      end


      add_file
         replace_if_existing = y
         replace_if_newer = n
         remove_if_modified = y
         name = TMV610.pdf
         translate = n
         destination = TMV610.pdf
         remove_empty_dirs = y
         is_shared = n
         remove_extraneous = n
         substitute_variables = n
         unix_owner = root
         unix_user_id = 0
         unix_group_id = 0
         create_dirs = y
         remote = n
         compute_crc = n
         verify_crc = n
         delta_compressible = d
         temporary = n
         is_signature = n
         compression_method = deflated
         rename_if_locked = n
      end

   end


   add_directory
      stop_on_failure = y
      add = y
      replace_if_existing = y
      replace_if_newer = n
      remove_if_modified = n
      location = $(source_dir)\WINDOWS
      name = LAP_JRE
      translate = n
      destination = $(CandleHome)\InstSpbs\WINDOWS\LAP_JRE
      descend_dirs = y
      remove_empty_dirs = y
      is_shared = n
      remove_extraneous = n
      substitute_variables = n
      unix_owner = root
      unix_user_id = 0
      unix_group_id = 0
      create_dirs = y
      remote = n
      compute_crc = n
      verify_crc = n
      delta_compressible = d
      temporary = n
      is_signature = n
      compression_method = deflated
      rename_if_locked = n
   end


   add_directory
      stop_on_failure = y
      add = y
      replace_if_existing = y
      replace_if_newer = n
      remove_if_modified = n
      location = $(source_dir)\WINDOWS
      name = LAP_LIC
      translate = n
      destination = $(CandleHome)\InstSpbs\WINDOWS\LAP_LIC
      descend_dirs = y
      remove_empty_dirs = y
      is_shared = n
      remove_extraneous = n
      substitute_variables = n
      unix_owner = root
      unix_user_id = 0
      unix_group_id = 0
      create_dirs = y
      remote = n
      compute_crc = n
      verify_crc = n
      delta_compressible = d
      temporary = n
      is_signature = n
      compression_method = deflated
      rename_if_locked = n
   end


   add_directory
      stop_on_failure = y
      add = y
      replace_if_existing = y
      replace_if_newer = n
      remove_if_modified = n
      location = $(source_dir)\WINDOWS
      name = LAP
      translate = n
      destination = $(CandleHome)\InstSpbs\WINDOWS\LAP
      descend_dirs = y
      remove_empty_dirs = y
      is_shared = n
      remove_extraneous = n
      substitute_variables = n
      unix_owner = root
      unix_user_id = 0
      unix_group_id = 0
      create_dirs = y
      remote = n
      compute_crc = n
      verify_crc = n
      delta_compressible = d
      temporary = n
      is_signature = n
      compression_method = deflated
      rename_if_locked = n
   end


   add_directory
      stop_on_failure = y
      add = y
      replace_if_existing = y
      replace_if_newer = n
      remove_if_modified = n
      location = $(source_dir)\WINDOWS
      name = InsGSKit
      translate = n
      destination = $(CandleHome)\InstSpbs\WINDOWS\InsGSKit
      descend_dirs = y
      remove_empty_dirs = y
      is_shared = n
      remove_extraneous = n
      substitute_variables = n
      unix_owner = root
      unix_user_id = 0
      unix_group_id = 0
      create_dirs = y
      remote = n
      compute_crc = n
      verify_crc = n
      delta_compressible = d
      temporary = n
      is_signature = n
      compression_method = deflated
      rename_if_locked = n
   end


   add_directory
      stop_on_failure = y
      add = y
      replace_if_existing = y
      replace_if_newer = n
      remove_if_modified = n
      location = $(source_dir)\WINDOWS
      name = VERFiles
      translate = n
      destination = $(CandleHome)\InstSpbs\WINDOWS\VERFiles
      descend_dirs = n
      remove_empty_dirs = y
      is_shared = n
      remove_extraneous = n
      substitute_variables = n
      unix_owner = root
      unix_user_id = 0
      unix_group_id = 0
      create_dirs = y
      remote = n
      compute_crc = n
      verify_crc = n
      delta_compressible = d
      temporary = n
      is_signature = n
      compression_method = deflated
      rename_if_locked = n

      add_file
         replace_if_existing = y
         replace_if_newer = n
         remove_if_modified = n
         name = KNTWICMA.VER
         translate = n
         destination = KNTWICMA.VER
         remove_empty_dirs = y
         is_shared = n
         remove_extraneous = n
         substitute_variables = n
         unix_owner = root
         unix_user_id = 0
         unix_group_id = 0
         create_dirs = y
         remote = n
         compute_crc = n
         verify_crc = n
         delta_compressible = d
         temporary = n
         is_signature = n
         compression_method = deflated
         rename_if_locked = n
      end


      add_file
         replace_if_existing = y
         replace_if_newer = n
         remove_if_modified = n
         name = KGLWICMA.VER
         translate = n
         destination = KGLWICMA.VER
         remove_empty_dirs = y
         is_shared = n
         remove_extraneous = n
         substitute_variables = n
         unix_owner = root
         unix_user_id = 0
         unix_group_id = 0
         create_dirs = y
         remote = n
         compute_crc = n
         verify_crc = n
         delta_compressible = d
         temporary = n
         is_signature = n
         compression_method = deflated
         rename_if_locked = n
      end


      add_file
         replace_if_existing = y
         replace_if_newer = n
         remove_if_modified = n
         name = KUIWICLI.VER
         translate = n
         destination = KUIWICLI.VER
         remove_empty_dirs = y
         is_shared = n
         remove_extraneous = n
         substitute_variables = n
```

```
            unix_owner = root
            unix_user_id = 0
            unix_group_id = 0
            create_dirs = y
            remote = n
            compute_crc = n
            verify_crc = n
            delta_compressible = d
            temporary = n
            is_signature = n
            compression_method = deflated
            rename_if_locked = n
         end


         add_file
            replace_if_existing = y
            replace_if_newer = n
            remove_if_modified = n
            name = KINWIINS.VER
            translate = n
            destination = KINWIINS.VER
            remove_empty_dirs = y
            is_shared = n
            remove_extraneous = n
            substitute_variables = n
            unix_owner = root
            unix_user_id = 0
            unix_group_id = 0
            create_dirs = y
            remote = n
            compute_crc = n
            verify_crc = n
            delta_compressible = d
            temporary = n
            is_signature = n
            compression_method = deflated
            rename_if_locked = n
         end

      end


      add_directory
         stop_on_failure = y
         add = y
         replace_if_existing = y
         replace_if_newer = n
         remove_if_modified = n
```

```
location = $(source_dir)\WINDOWS
name = CLEVEL
translate = n
destination = $(CandleHome)\InstSpbs\WINDOWS\CLEVEL
descend_dirs = n
remove_empty_dirs = y
is_shared = n
remove_extraneous = n
substitute_variables = n
unix_owner = root
unix_user_id = 0
unix_group_id = 0
create_dirs = y
remote = n
compute_crc = n
verify_crc = n
delta_compressible = d
temporary = n
is_signature = n
compression_method = deflated
rename_if_locked = n

add_file
   replace_if_existing = y
   replace_if_newer = n
   remove_if_modified = n
   name = KNTWICMA.LVL
   translate = n
   destination = KNTWICMA.LVL
   remove_empty_dirs = y
   is_shared = n
   remove_extraneous = n
   substitute_variables = n
   unix_owner = root
   unix_user_id = 0
   unix_group_id = 0
   create_dirs = y
   remote = n
   compute_crc = n
   verify_crc = n
   delta_compressible = d
   temporary = n
   is_signature = n
   compression_method = deflated
   rename_if_locked = n
end


add_file
```

```
   replace_if_existing = y
   replace_if_newer = n
   remove_if_modified = n
   name = KGLWICMA.LVL
   translate = n
   destination = KGLWICMA.LVL
   remove_empty_dirs = y
   is_shared = n
   remove_extraneous = n
   substitute_variables = n
   unix_owner = root
   unix_user_id = 0
   unix_group_id = 0
   create_dirs = y
   remote = n
   compute_crc = n
   verify_crc = n
   delta_compressible = d
   temporary = n
   is_signature = n
   compression_method = deflated
   rename_if_locked = n
end


add_file
   replace_if_existing = y
   replace_if_newer = n
   remove_if_modified = n
   name = KUIWICLI.LVL
   translate = n
   destination = KUIWICLI.LVL
   remove_empty_dirs = y
   is_shared = n
   remove_extraneous = n
   substitute_variables = n
   unix_owner = root
   unix_user_id = 0
   unix_group_id = 0
   create_dirs = y
   remote = n
   compute_crc = n
   verify_crc = n
   delta_compressible = d
   temporary = n
   is_signature = n
   compression_method = deflated
   rename_if_locked = n
end
```

```
add_file
   replace_if_existing = y
   replace_if_newer = n
   remove_if_modified = n
   name = KinCLevel.exe
   translate = n
   destination = KinCLevel.exe
   remove_empty_dirs = y
   is_shared = n
   remove_extraneous = n
   substitute_variables = n
   unix_owner = root
   unix_user_id = 0
   unix_group_id = 0
   create_dirs = y
   remote = n
   compute_crc = n
   verify_crc = n
   delta_compressible = d
   temporary = n
   is_signature = n
   compression_method = deflated
   rename_if_locked = n
end


add_file
   replace_if_existing = y
   replace_if_newer = n
   remove_if_modified = n
   name = MFC42.DLL
   translate = n
   destination = MFC42.DLL
   remove_empty_dirs = y
   is_shared = n
   remove_extraneous = n
   substitute_variables = n
   unix_owner = root
   unix_user_id = 0
   unix_group_id = 0
   create_dirs = y
   remote = n
   compute_crc = n
   verify_crc = n
   delta_compressible = d
   temporary = n
   is_signature = n
```

```
               compression_method = deflated
               rename_if_locked = n
            end


            add_file
               replace_if_existing = y
               replace_if_newer = n
               remove_if_modified = n
               name = msvcrt.dll
               translate = n
               destination = msvcrt.dll
               remove_empty_dirs = y
               is_shared = n
               remove_extraneous = n
               substitute_variables = n
               unix_owner = root
               unix_user_id = 0
               unix_group_id = 0
               create_dirs = y
               remote = n
               compute_crc = n
               verify_crc = n
               delta_compressible = d
               temporary = n
               is_signature = n
               compression_method = deflated
               rename_if_locked = n
            end

         end

      end


      execute_user_program
         caption = "Silent Install"
         transactional = n

         during_install
            path = cmd
            arguments = '/c start /wait $(CandleHome)\InstSpbs\Windows\setup.exe
   /w /z"/sf$(CandleHome)\InstSpbs\silent_kntcma_install.txt" /s
   /f2"$(temp_dir)\silent_kntcma_install.log"'
            inhibit_parsing = y
            working_dir = $(CandleHome)\InstSpbs\Windows
            timeout = -1
            unix_user_id = 0
            unix_group_id = 0
            user_input_required = n
            output_file = $(temp_dir)\kntwicma.out
            error_file = $(temp_dir)\kntwicma.out
            output_file_append = y
            error_file_append = y
            reporting_stdout_on_server = n
            reporting_stderr_on_server = n
            max_stdout_size = 10000
            max_stderr_size = 10000
            bootable = n
            retry = 1

            exit_codes
               success = 0,0
               failure = 1,65535
            end

         end


         during_remove
            path = cmd
            arguments = '/c start /wait $(CandleHome)\InstSpbs\Windows\setup.exe
   /w /z"/sf$(CandleHome)\InstSpbs\silent_kntcma_uninstall.txt" /s
   /f2"$(temp_dir)\silent_kntcma_uninstall.log"'
            inhibit_parsing = y
            working_dir = $(CandleHome)\InstSpbs\Windows
            timeout = -1
            unix_user_id = 0
            unix_group_id = 0
            user_input_required = n
            output_file = $(temp_dir)\kntwicma.out
            error_file = $(temp_dir)\kntwicma.out
            output_file_append = y
            error_file_append = y
            reporting_stdout_on_server = n
            reporting_stderr_on_server = n
            max_stdout_size = 10000
            max_stderr_size = 10000
            bootable = n
            retry = 1

            exit_codes
               success = 0,0
               failure = 1,65535
            end

         end
```

```
end


execute_user_program
   caption = "Check Silent Install Log"
   transactional = n

   during_install
      path = '"$(temp_dir)\checkIsLog.bat"'
      arguments = silent_kntcma_install.log
      inhibit_parsing = y
      working_dir = $(temp_dir)
      timeout = -1
      unix_user_id = 0
      unix_group_id = 0
      user_input_required = n
      output_file = $(temp_dir)\kntwicma.out
      error_file = $(temp_dir)\kntwicma.out
      output_file_append = y
      error_file_append = y
      reporting_stdout_on_server = n
      reporting_stderr_on_server = n
      max_stdout_size = 10000
      max_stderr_size = 10000
      bootable = n
      retry = 1

      exit_codes
         success = 0,0
         failure = 1,65535
      end


      corequisite_files

         add_file
            replace_if_existing = y
            replace_if_newer = n
            remove_if_modified = y
            name = $(silentfiles_dir)\checkIsLog.bat
            translate = n
            destination = $(temp_dir)\checkIsLog.bat
            unix_owner = root
            unix_user_id = 0
            unix_group_id = 0
            compression_method = deflated
            rename_if_locked = n
         end
```

```
            end

         end

      end

   end
```

The Universal Agent SPD for AIX

Example A-2 shows the Universal Agent SPD for AIX (UMAIX.SPD).

Example: A-2 The Universal Agent SPD (UMAIX.SPD)

```
"TIVOLI Software Package v4.2.2 - SPDF"

package
   name = UMAIX
   title = "Universal Agent"
   version = 6.1.0
   web_view_mode = hidden
   undoable = o
   committable = o
   history_reset = n
   save_default_variables = n
   creation_time = "2005-10-12 17:43:20"
   last_modification_time = "2005-10-12 17:46:09"

   default_variables
      CandleHome = /opt/IBM/ITM
      source_dir = C:\cm_inst\ITM\ITM6_SPBs\images\S1\unix
      silentfiles_dir = C:\cm_inst\ITM\ITM6_SPBs\images\SilentFiles\Unix
   end

   move_removing_host = y
   no_check_source_host = y
   lenient_distribution = n
   default_operation = install
   server_mode = all
   operation_mode = not_transactional
   post_notice = n
   before_as_uid = 0
   skip_non_zero = n
   after_as_uid = 0
   no_chk_on_rm = y
   versioning_type = swd
   package_type = refresh
```

```
dependency = "$(installed_software) >= UXAIX.6.1.0"
sharing_control = none
stop_on_failure = y

generic_container
   caption = "Silent Files"
   stop_on_failure = y
   condition = "$(os_name)  == AIX"

   add_directory
      stop_on_failure = y
      add = y
      replace_if_existing = y
      replace_if_newer = n
      remove_if_modified = n
      location = $(silentfiles_dir)
      translate = n
      destination = $(CandleHome)\InstSpbs
      descend_dirs = n
      remove_empty_dirs = y
      is_shared = n
      remove_extraneous = n
      substitute_variables = y
      unix_owner = root
      unix_user_id = 0
      unix_group_id = 0
      create_dirs = y
      remote = n
      compute_crc = n
      verify_crc = n
      delta_compressible = d
      temporary = n
      is_signature = n
      compression_method = deflated
      rename_if_locked = n

      add_file
         replace_if_existing = y
         replace_if_newer = n
         remove_if_modified = y
         name = silent_um_install.txt
         translate = n
         destination = silent_um_install.txt
         remove_empty_dirs = y
         is_shared = n
         remove_extraneous = n
         substitute_variables = y
         unix_owner = root
         unix_user_id = 0
```

```
            unix_group_id = 0
            create_dirs = y
            remote = n
            compute_crc = n
            verify_crc = n
            delta_compressible = d
            temporary = n
            is_signature = n
            compression_method = deflated
            rename_if_locked = n
         end

      end

   end


   execute_user_program
      caption = "Stop Running Agents"
      condition = "$(os_name)  == AIX"
      transactional = n

      during_install
         path = $(CandleHome)\bin\itmcmd
         arguments = "agent stop all"
         inhibit_parsing = n
         working_dir = $(CandleHome)\InstSpbs
         timeout = -1
         unix_user_id = 0
         unix_group_id = 0
         user_input_required = n
         output_file = $(temp_dir)\umaix.out
         error_file = $(temp_dir)\umaix.out
         output_file_append = y
         error_file_append = y
         reporting_stdout_on_server = n
         reporting_stderr_on_server = n
         max_stdout_size = 10000
         max_stderr_size = 10000
         bootable = n
         retry = 1

         exit_codes
            success = 0,65535
         end

      end
      during_remove
         path = $(CandleHome)\bin\itmcmd
         arguments = "agent start all"
         inhibit_parsing = n
         working_dir = $(CandleHome)\InstSpbs
         timeout = -1
         unix_user_id = 0
         unix_group_id = 0
         user_input_required = n
         output_file = $(temp_dir)\umaix.out
         error_file = $(temp_dir)\umaix.out
         output_file_append = y
         error_file_append = y
         reporting_stdout_on_server = n
         reporting_stderr_on_server = n
         max_stdout_size = 10000
         max_stderr_size = 10000
         bootable = n
         retry = 1

         exit_codes
            success = 0,0
            failure = 1,65535
         end

      end

   end


   execute_user_program
      caption = "Silent Install Universal Agent"
      condition = "$(os_name)  == AIX"
      transactional = n

      during_install
         path = $(CandleHome)\InstSpbs\UNIX\install.sh
         arguments = "-q -h $(CandleHome) -p
$(CandleHome)/InstSpbs/silent_um_install.txt"
         inhibit_parsing = y
         working_dir = $(CandleHome)\InstSpbs\UNIX
         timeout = -1
         unix_user_id = 0
         unix_group_id = 0
         user_input_required = n
         output_file = $(temp_dir)\umaix.out
         error_file = $(temp_dir)\umaix.out
         output_file_append = y
         error_file_append = y
```

```
reporting_stdout_on_server = n
reporting_stderr_on_server = n
max_stdout_size = 10000
max_stderr_size = 10000
bootable = n
retry = 1

exit_codes
   success = 0,0
   failure = 1,65535
end


corequisite_files

   add_directory
      replace_if_existing = y
      replace_if_newer = n
      remove_if_modified = y
      location = $(source_dir)
      destination = $(CandleHome)\InstSpbs\UNIX
      descend_dirs = n
      unix_owner = root
      unix_user_id = 0
      unix_group_id = 0
      compression_method = deflated
      rename_if_locked = n

      add_file
         replace_if_existing = y
         replace_if_newer = y
         remove_if_modified = y
         name = install.sh
         translate = n
         destination = install.sh
         unix_attributes = rwx,rx,
         unix_owner = root
         unix_user_id = 0
         unix_group_id = 0
         compression_method = deflated
         rename_if_locked = n
      end

   end


   add_directory
      replace_if_existing = y
      replace_if_newer = n
```

```
remove_if_modified = n
location = $(source_dir)
name = unix
destination = $(CandleHome)\InstSpbs\UNIX\unix
descend_dirs = n
unix_owner = root
unix_user_id = 0
unix_group_id = 0
compression_method = deflated
rename_if_locked = n

add_file
   replace_if_existing = y
   replace_if_newer = n
   remove_if_modified = y
   name = ufaix513.jar
   translate = n
   destination = ufaix513.jar
   unix_owner = root
   unix_user_id = 0
   unix_group_id = 0
   compression_method = deflated
   rename_if_locked = n
end


add_file
   replace_if_existing = y
   replace_if_newer = n
   remove_if_modified = y
   name = ufaix516.jar
   translate = n
   destination = ufaix516.jar
   unix_owner = root
   unix_user_id = 0
   unix_group_id = 0
   compression_method = deflated
   rename_if_locked = n
end


add_file
   replace_if_existing = y
   replace_if_newer = n
   remove_if_modified = y
   name = umaix513.jar
   translate = n
   destination = umaix513.jar
   unix_owner = root
```

```
            unix_user_id = 0
            unix_group_id = 0
            compression_method = deflated
            rename_if_locked = n
         end


         add_file
            replace_if_existing = y
            replace_if_newer = n
            remove_if_modified = y
            name = umaix516.jar
            translate = n
            destination = umaix516.jar
            unix_owner = root
            unix_user_id = 0
            unix_group_id = 0
            compression_method = deflated
            rename_if_locked = n
         end

      end

   end

end


during_remove
   path = $(temp_dir)\unAgSpb\spb_uninstall.sh
   arguments = "$(CandleHome) um $(os_name)"
   inhibit_parsing = n
   working_dir = $(temp_dir)\unAgSpb
   timeout = -1
   unix_user_id = 0
   unix_group_id = 0
   user_input_required = n
   output_file = $(temp_dir)\umaix.out
   error_file = $(temp_dir)\umaix.out
   output_file_append = y
   error_file_append = y
   reporting_stdout_on_server = n
   reporting_stderr_on_server = n
   max_stdout_size = 10000
   max_stderr_size = 10000
   bootable = n
   retry = 1

   exit_codes
```

```
            success = 0,0
            warning = 1,1
            failure = 2,65535
         end


         corequisite_files

            add_directory
               replace_if_existing = y
               replace_if_newer = n
               remove_if_modified = y
               location = $(silentfiles_dir)
               destination = $(temp_dir)\unAgSpb
               descend_dirs = n
               unix_owner = root
               unix_user_id = 0
               unix_group_id = 0
               compression_method = deflated
               rename_if_locked = n

               add_file
                  replace_if_existing = y
                  replace_if_newer = n
                  remove_if_modified = y
                  name = spb_uninstall.sh
                  translate = n
                  destination = spb_uninstall.sh
                  unix_attributes = rwx,rx,
                  unix_owner = root
                  unix_user_id = 0
                  unix_group_id = 0
                  compression_method = deflated
                  rename_if_locked = n
               end

            end

         end

      end

   end


   execute_user_program
      caption = "Start Agents"
      condition = "$(os_name)  == AIX"
      transactional = n
```

```
during_install
   path = $(CandleHome)\bin\itmcmd
   arguments = "agent start all"
   inhibit_parsing = n
   working_dir = $(CandleHome)\InstSpbs
   timeout = -1
   unix_user_id = 0
   unix_group_id = 0
   user_input_required = n
   output_file = $(temp_dir)\umaix.out
   error_file = $(temp_dir)\umaix.out
   output_file_append = y
   error_file_append = y
   reporting_stdout_on_server = n
   reporting_stderr_on_server = n
   max_stdout_size = 10000
   max_stderr_size = 10000
   bootable = n
   retry = 1

   exit_codes
      success = 0,0
      failure = 1,65535
   end

end


during_remove
   path = $(CandleHome)\bin\itmcmd
   arguments = "agent stop all"
   inhibit_parsing = n
   working_dir = $(CandleHome)\InstSpbs
   timeout = -1
   unix_user_id = 0
   unix_group_id = 0
   user_input_required = n
   output_file = $(temp_dir)\umaix.out
   error_file = $(temp_dir)\umaix.out
   output_file_append = y
   error_file_append = y
   reporting_stdout_on_server = n
   reporting_stderr_on_server = n
   max_stdout_size = 10000
   max_stderr_size = 10000
   bootable = n
   retry = 1
```

```
            exit_codes
               success = 0,0
               failure = 1,65535
            end

         end

      end

   end
```

The AIX OS Agent SPD

Example A-3 shows the AIX OS Agent SPD (UXAIX.SPD).

Example: A-3 The AIX OS Agent SPD (UXAIX.SPD)

```
"TIVOLI Software Package v4.2.2 - SPDF"

package
   name = UXAIX
   title = "Monitoring Agent for UNIX OS"
   version = 6.1.0
   web_view_mode = hidden
   undoable = o
   committable = o
   history_reset = n
   save_default_variables = n
   creation_time = "2005-10-20 11:54:44"
   last_modification_time = "2005-10-20 12:00:45"

   default_variables
      CandleEncryptionKey = IBMTivoliMonitoringEncryptionKey
      SNALUName = LUNAME
      CandleHome = /opt/IBM/ITM
      NetworkProtocol = ip.pipe
      PortNumber = 1918
      source_dir = C:\cm_inst\ITM\ITM6_SPBs\images\S1\unix
      TEMS_Hostname = $(hostname)
      SNANetName = CANDLE
      IPSPipePortNumber = 3660
      SNALogMode = LOGMODE
      silentfiles_dir = C:\cm_inst\ITM\ITM6_SPBs\images\SilentFiles\Unix
      IPPipePortNumber = 1918
   end

   move_removing_host = y
```

```
no_check_source_host = y
lenient_distribution = n
default_operation = install
server_mode = all
operation_mode = not_transactional
post_notice = n
before_as_uid = 0
skip_non_zero = n
after_as_uid = 0
no_chk_on_rm = y
versioning_type = swd
package_type = refresh
sharing_control = none
stop_on_failure = y

generic_container
   caption = "Silent Files"
   stop_on_failure = y
   condition = "$(os_name)  == AIX"

   add_directory
      stop_on_failure = y
      add = y
      replace_if_existing = y
      replace_if_newer = n
      remove_if_modified = n
      location = $(silentfiles_dir)
      translate = n
      destination = $(CandleHome)\InstSpbs
      descend_dirs = n
      remove_empty_dirs = y
      is_shared = n
      remove_extraneous = n
      substitute_variables = y
      unix_owner = root
      unix_user_id = 0
      unix_group_id = 0
      create_dirs = y
      remote = n
      compute_crc = n
      verify_crc = n
      delta_compressible = d
      temporary = n
      is_signature = n
      compression_method = deflated
      rename_if_locked = n

      add_file
         replace_if_existing = y
```

```
        replace_if_newer = y
        remove_if_modified = y
        name = silent_ux_install.txt
        translate = n
        destination = silent_ux_install.txt
        remove_empty_dirs = y
        is_shared = n
        remove_extraneous = n
        substitute_variables = y
        unix_owner = root
        unix_user_id = 0
        unix_group_id = 0
        create_dirs = y
        remote = n
        compute_crc = n
        verify_crc = n
        delta_compressible = d
        temporary = n
        is_signature = n
        compression_method = deflated
        rename_if_locked = n
    end


    add_file
        replace_if_existing = y
        replace_if_newer = n
        remove_if_modified = y
        name = silent_config.txt
        translate = n
        destination = silent_config.txt
        remove_empty_dirs = y
        is_shared = n
        remove_extraneous = n
        substitute_variables = y
        unix_owner = root
        unix_user_id = 0
        unix_group_id = 0
        create_dirs = y
        remote = n
        compute_crc = n
        verify_crc = n
        delta_compressible = d
        temporary = n
        is_signature = n
        compression_method = deflated
        rename_if_locked = n
    end
```

```
      end

   end


   execute_user_program
      caption = "Silent Install Monitoring Agent for UNIX OS"
      condition = "$(os_name)  == AIX"
      transactional = n

      during_install
         path = $(CandleHome)\InstSpbs\UNIX\install.sh
         arguments = "-q -h $(CandleHome) -p
$(CandleHome)/InstSpbs/silent_ux_install.txt"
         inhibit_parsing = y
         working_dir = $(CandleHome)\InstSpbs\UNIX
         timeout = -1
         unix_user_id = 0
         unix_group_id = 0
         user_input_required = n
         output_file = $(temp_dir)\uxaix.out
         error_file = $(temp_dir)\uxaix.out
         output_file_append = y
         error_file_append = y
         reporting_stdout_on_server = n
         reporting_stderr_on_server = n
         max_stdout_size = 10000
         max_stderr_size = 10000
         bootable = n
         retry = 1

         exit_codes
            success = 0,0
            failure = 1,65535
         end


         corequisite_files

            add_directory
               replace_if_existing = y
               replace_if_newer = n
               remove_if_modified = y
               location = $(source_dir)
               destination = $(CandleHome)\InstSpbs\UNIX
               descend_dirs = n
               unix_owner = root
               unix_user_id = 0
               unix_group_id = 0
               compression_method = deflated
               rename_if_locked = n

               add_file
                  replace_if_existing = y
                  replace_if_newer = y
                  remove_if_modified = y
                  name = install.sh
                  translate = n
                  destination = install.sh
                  unix_attributes = rwx,rx,
                  unix_owner = root
                  unix_user_id = 0
                  unix_group_id = 0
                  compression_method = deflated
                  rename_if_locked = n
               end

            end


            add_directory
               replace_if_existing = y
               replace_if_newer = n
               remove_if_modified = n
               location = $(source_dir)
               name = unix
               destination = $(CandleHome)\InstSpbs\UNIX\unix
               descend_dirs = n
               unix_owner = root
               unix_user_id = 0
               unix_group_id = 0
               compression_method = deflated
               rename_if_locked = n

               add_file
                  replace_if_existing = y
                  replace_if_newer = n
                  remove_if_modified = y
                  name = axaix513.jar
                  translate = n
                  destination = axaix513.jar
                  unix_owner = root
                  unix_user_id = 0
                  unix_group_id = 0
                  compression_method = deflated
                  rename_if_locked = n
               end
```

```
add_file
   replace_if_existing = y
   replace_if_newer = n
   remove_if_modified = y
   name = axaix516.jar
   translate = n
   destination = axaix516.jar
   unix_owner = root
   unix_user_id = 0
   unix_group_id = 0
   compression_method = deflated
   rename_if_locked = n
end


add_file
   replace_if_existing = y
   replace_if_newer = n
   remove_if_modified = y
   name = cienv.tar
   translate = n
   destination = cienv.tar
   unix_owner = root
   unix_user_id = 0
   unix_group_id = 0
   compression_method = deflated
   rename_if_locked = n
end


add_file
   replace_if_existing = y
   replace_if_newer = n
   remove_if_modified = y
   name = cienv1.tar
   translate = n
   destination = cienv1.tar
   unix_owner = root
   unix_user_id = 0
   unix_group_id = 0
   compression_method = deflated
   rename_if_locked = n
end


add_file
   replace_if_existing = y
   replace_if_newer = n
   remove_if_modified = y
   name = jraix513.ctp
   translate = n
   destination = jraix513.ctp
   unix_owner = root
   unix_user_id = 0
   unix_group_id = 0
   compression_method = deflated
   rename_if_locked = n
end


add_file
   replace_if_existing = y
   replace_if_newer = n
   remove_if_modified = y
   name = jraix516.ctp
   translate = n
   destination = jraix516.ctp
   unix_owner = root
   unix_user_id = 0
   unix_group_id = 0
   compression_method = deflated
   rename_if_locked = n
end


add_file
   replace_if_existing = y
   replace_if_newer = n
   remove_if_modified = y
   name = uiaix513.jar
   translate = n
   destination = uiaix513.jar
   unix_owner = root
   unix_user_id = 0
   unix_group_id = 0
   compression_method = deflated
   rename_if_locked = n
end


add_file
   replace_if_existing = y
   replace_if_newer = n
   remove_if_modified = y
   name = uiaix516.jar
   translate = n
   destination = uiaix516.jar
```

```
         unix_owner = root
         unix_user_id = 0
         unix_group_id = 0
         compression_method = deflated
         rename_if_locked = n
      end


      add_file
         replace_if_existing = y
         replace_if_newer = n
         remove_if_modified = y
         name = uxaix513.jar
         translate = n
         destination = uxaix513.jar
         unix_owner = root
         unix_user_id = 0
         unix_group_id = 0
         compression_method = deflated
         rename_if_locked = n
      end


      add_file
         replace_if_existing = y
         replace_if_newer = n
         remove_if_modified = y
         name = uxaix516.jar
         translate = n
         destination = uxaix516.jar
         unix_owner = root
         unix_user_id = 0
         unix_group_id = 0
         compression_method = deflated
         rename_if_locked = n
      end

   end


   add_directory
      replace_if_existing = y
      replace_if_newer = n
      remove_if_modified = y
      location = $(source_dir)\unix
      name = GSKit
      destination = $(CandleHome)\InstSpbs\UNIX\unix\GSKit
      descend_dirs = n
      unix_owner = root
      unix_user_id = 0
      unix_group_id = 0
      compression_method = deflated
      rename_if_locked = n

      add_directory
         stop_on_failure = y
         replace_if_existing = y
         replace_if_newer = n
         remove_if_modified = y
         name = aix516
         destination = aix516
         descend_dirs = y
         unix_owner = root
         unix_user_id = 0
         unix_group_id = 0
         compression_method = deflated
         rename_if_locked = n
      end


      add_directory
         stop_on_failure = y
         replace_if_existing = y
         replace_if_newer = n
         remove_if_modified = y
         name = aix513
         destination = aix513
         descend_dirs = y
         unix_owner = root
         unix_user_id = 0
         unix_group_id = 0
         compression_method = deflated
         rename_if_locked = n
      end

   end


   add_directory
      replace_if_existing = y
      replace_if_newer = n
      remove_if_modified = y
      location = $(source_dir)\unix
      name = LAP
      destination = $(CandleHome)\InstSpbs\UNIX\unix\LAP
      descend_dirs = y
      unix_owner = root
      unix_user_id = 0
```

```
            unix_group_id = 0
            compression_method = deflated
            rename_if_locked = n
         end

      end

   end


   during_remove
      path = $(temp_dir)\unAgSpb\spb_uninstall.sh
      arguments = "$(CandleHome) ux $(os_name)"
      inhibit_parsing = n
      working_dir = $(temp_dir)\unAgSpb
      timeout = -1
      unix_user_id = 0
      unix_group_id = 0
      user_input_required = n
      output_file = $(temp_dir)\uxaix.out
      error_file = $(temp_dir)\uxaix.out
      output_file_append = y
      error_file_append = y
      reporting_stdout_on_server = n
      reporting_stderr_on_server = n
      max_stdout_size = 10000
      max_stderr_size = 10000
      bootable = n
      retry = 1

      exit_codes
         success = 0,0
         warning = 1,1
         failure = 2,65535
      end


      corequisite_files

         add_directory
            replace_if_existing = y
            replace_if_newer = n
            remove_if_modified = y
            location = $(silentfiles_dir)
            destination = $(temp_dir)\unAgSpb
            descend_dirs = n
            unix_owner = root
            unix_user_id = 0
            unix_group_id = 0
```

```
                compression_method = deflated
                rename_if_locked = n

                add_file
                   replace_if_existing = y
                   replace_if_newer = n
                   remove_if_modified = y
                   name = spb_uninstall.sh
                   translate = n
                   destination = spb_uninstall.sh
                   unix_attributes = rwx,rx,
                   unix_owner = root
                   unix_user_id = 0
                   unix_group_id = 0
                   compression_method = deflated
                   rename_if_locked = n
                end

             end

          end

       end

    end


    execute_user_program
       caption = "Configure Monitoring Agent for UNIX OS"
       condition = "$(os_name)  == AIX"
       transactional = n

       during_install
          path = $(CandleHome)\bin\itmcmd
          arguments = "config -A -p $(CandleHome)/InstSpbs/silent_config.txt
 ux"
          inhibit_parsing = y
          working_dir = $(CandleHome)\InstSpbs
          timeout = -1
          unix_user_id = 0
          unix_group_id = 0
          user_input_required = n
          output_file = $(temp_dir)\uxaix.out
          error_file = $(temp_dir)\uxaix.out
          output_file_append = y
          error_file_append = y
          reporting_stdout_on_server = n
          reporting_stderr_on_server = n
          max_stdout_size = 10000
```

```
         max_stderr_size = 10000
         bootable = n
         retry = 1

         exit_codes
            success = 0,0
            failure = 1,65535
         end

      end

   end


   execute_user_program
      caption = "Start Agents"
      condition = "$(os_name)  == AIX"
      transactional = n

      during_install
         path = $(CandleHome)\bin\itmcmd
         arguments = "agent start all"
         inhibit_parsing = n
         working_dir = $(CandleHome)\InstSpbs
         timeout = -1
         unix_user_id = 0
         unix_group_id = 0
         user_input_required = n
         output_file = $(temp_dir)\uxaix.out
         error_file = $(temp_dir)\uxaix.out
         output_file_append = y
         error_file_append = y
         reporting_stdout_on_server = n
         reporting_stderr_on_server = n
         max_stdout_size = 10000
         max_stderr_size = 10000
         bootable = n
         retry = 1

         exit_codes
            success = 0,0
            failure = 1,65535
         end

      end


      during_remove
         path = $(CandleHome)\bin\itmcmd
         arguments = "agent stop all"
         inhibit_parsing = n
         working_dir = $(CandleHome)\InstSpbs
         timeout = -1
         unix_user_id = 0
         unix_group_id = 0
         user_input_required = n
         output_file = $(temp_dir)\uxaix.out
         error_file = $(temp_dir)\uxaix.out
         output_file_append = y
         error_file_append = y
         reporting_stdout_on_server = n
         reporting_stderr_on_server = n
         max_stdout_size = 10000
         max_stderr_size = 10000
         bootable = n
         retry = 1

         exit_codes
            success = 0,0
            failure = 1,65535
         end

      end

   end

end
```

B

Additional material

This book refers to additional material that can be downloaded from the Internet as described below.

Locating the Web material

The Web material associated with this book is available in softcopy on the Internet from the IBM Redbooks Web server. Point your Web browser to:

`ftp://www.redbooks.ibm.com/redbooks/SG247143`

Alternatively, you can go to the IBM Redbooks Web site at:

ibm.com/redbooks

Select the **Additional materials** and open the directory that corresponds with the redbook form number, SG247143.

Using the Web material

The additional Web material that accompanies this book includes this file:

File name	*Description*
SG247143.zip	Zipped SPD packages for TEMA installation using IBM Tivoli Configuration Manager

System requirements for downloading the Web material

The following system configuration is recommended:

Hard disk space:	20 MB minimum
Operating System:	Windows, Linux, or UNIX

How to use the Web material

Create a subdirectory (folder) on your workstation, and unzip the contents of the Web material zip file into this folder.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this Redbook.

IBM Redbooks

For information about ordering these publications, see "How to get IBM Redbooks" on page 807. Note that some of the documents referenced here may be available in softcopy only.

- *Certification Study Guide: IBM Tivoli Configuration Manager (ITCM) Version 4.2.2*, SG24-6691
- *Deployment Guide Series: IBM Tivoli Monitoring V 6.1*, SG24-7188
- *IBM Tivoli OMEGAMON V3.1.0 Deep Dive on z/OS*, SG24-7155 (draft available late December 2005)

Other publications

These publications are also relevant as further information sources:

- *IBM Tivoli Monitoring Quick Start Guide*, SC32-1802
- *Introducing IBM Tivoli Monitoring, V6.1.0,* GI11-4071
- *IBM Tivoli Monitoring User's Guide*, SC32-9409
- *IBM Tivoli Monitoring Administrator's Guide*, SC32-9408
- *IBM Tivoli Monitoring: Upgrading from Tivoli Distributed Monitoring*, SC32-9462
- *IBM Tivoli Monitoring: IBM Tivoli Monitoring 5.x Endpoint Agent User's Guide*, SC32-9490
- *IBM Tivoli Monitoring Installation and Setup Guide*, GC32-9407
- *IBM Tivoli Event Integration Facility Reference, Version 3.9*, SC32-1241
- *Tivoli Management Framework Reference Manual, Version 4.1.1*, SC32-0806
- *IBM Tivoli Configuration Manager User's Guide for Software Distribution*, SC23-4711

- *IBM Tivoli Configuration Manager Reference Manual for Software Distribution*, SC23-4712
- *IBM Tivoli Monitoring Problem Determination Guide,* GC32-9458
- *IBM Tivoli Universal Agent User's Guide,* SC32-9459

Online resources

These Web sites and URLs are also relevant as further information sources:

- DB2 Fix Pack Web site

  `http://www.ibm.com/software/data/db2/udb/support/downloadv8.html`
- Microsoft SQL Server downloads site

  `http://www.microsoft.com/sql/downloads/default.asp`
- Oracle ODBC drivers Web site

  `http://www.oracle.com/technology/software/tech/windows/odbc/htdocs/utilsoft.html`
- Business Objects Web site

  `http://www.businessobjects.com`
- IBM Java JRE Web site

  `http://www.ibm.com/developerworks/java/jdk`
- IBM Tivoli Support patch download site

  `ftp://ftp.software.ibm.com/software/tivoli_support/patches/`
- IBM Tivoli Monitoring 6.1 for Operators course description

  `http://www.ibm.com/software/tivoli/education/Q956702H03058N69.html`
- IBM Tivoli Monitoring 6.1 for Administrators course description

  `http://www.ibm.com/software/tivoli/education/G260590S46948M98.html`
- Wikipedia: Database normalization

  `http://en.wikipedia.org/wiki/Database_normalization`

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications, and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

IBM

Redbooks

Getting Started with IBM Tivoli Monitoring 6.1 on Distributed Environments

(1.5” spine)
1.5”<-> 1.998”
789 <->1051 pages

Getting Started with IBM Tivoli Monitoring 6.1 on Distributed Environments

ITM 6.1 implementation, best practices, and troubleshooting

DM 3.7 migration and ITM 5.1 coexistence scenarios

Insider's guide to ITM 6.1

The IBM Tivoli Monitoring Version 6.1 solution is the next generation of the IBM Tivoli family of products that help monitor and manage critical hardware and software in distributed environments. IBM Tivoli Monitoring 6.1 has emerged from the best of the IBM Tivoli Monitoring Version 5 and OMEGAMON technologies. Integration of these products creates a unique and comprehensive solution to monitor and manage both z/OS and distributed environments.

IBM Tivoli Monitoring 6.1 is easily customizable and provides real-time and historical data that enables you to quickly diagnose and solve issues with the new GUI via the IBM Tivoli Enterprise Portal component. This common, flexible, and easy-to-use browser interface helps users to quickly isolate and resolve potential performance problems.

This IBM Redbook covers planning, architecture, tuning, implementation, and troubleshooting of IBM Tivoli Monitoring 6.1. In addition, we offer scenarios for migration from Distributed Monitoring 3.7, and IBM Tivoli Monitoring 5.X coexistence with IBM Tivoli Monitoring 6.1.

This book is targeted for IT specialists who will be working on new IBM Tivoli Monitoring 6.1 installations on distributed environments or implementing migration from Distributed Monitoring 3.7 or IBM Tivoli Monitoring 5.X coexistence.

SG24-7143-00 ISBN 0738494186