Tivoli. software

IBM

Deployment Guide Series

IBM Tivoli Monitoring Express Version 6.1

- Provides a step-by-step deployment guide for IBM Tivoli Monitoring Express
- Discusses best practices for a deployment plan
- Describes architecture and planning considerations

Vasfi Gucer
Frederic Marsaud
Haakan Gradin
John Willis

Redbooks

ibm.com/redbooks

International Technical Support Organization

Deployment Guide Series: IBM Tivoli Monitoring Express Version 6.1

May 2006

SG24-7217-00

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

First Edition (May 2006)

This edition applies to IBM Tivoli Monitoring Express Version 6.1.

Contents

Notices

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

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

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

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

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

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

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

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

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

Trademarks

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

AF/OPERATOR®
AF/REMOTE®
AIX 5L™
AIX®
Candle®
DB2 Universal Database™
DB2®
i5/OS®
IBM®
NetView®
OMEGAMON®
OS/400®
Passport Advantage®
Redbooks (logo) ™
Redbooks™
Tivoli Enterprise Console®
Tivoli Enterprise™
Tivoli®
WebSphere®
z/OS®
zSeries®

The following terms are trademarks of other companies:

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

BizTalk, JScript, Microsoft, MS-DOS, SharePoint, Windows NT, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Pentium, 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

IBM® Tivoli® Monitoring Express Version 6.1 is a powerful, affordable, and easy-to-use availability management solution designed to help small to mid-sized companies manage IT infrastructures. It offers the ability to manage bottlenecks, performance impacts, and outages across heterogeneous environments from a single, centralized portal.

IBM Tivoli Monitoring Express V6.1 is easy to install, easy to deploy, and easy to use, providing rapid time to value. It provides real-time and historical data that enables you to quickly diagnose and solve issues with the new GUI through the IBM Tivoli Enterprise™ Portal component.

This IBM Redbook presents a deployment guide for IBM Tivoli Monitoring Express V6.1. We describe planning, installing, and troubleshooting IBM Tivoli Monitoring Express V6.1. In addition, we provide some case studies that you can use as part of a proof of concept or a customer demonstration.

The target audience for this book is IT specialists working on new IBM Tivoli Monitoring Express V6.1 installations.

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.

Vasfi Gucer is an IBM Certified Consultant IT Specialist at the ITSO, Austin Center. He started his IBM career as a Network Specialist in 1989. He has more than 16 years of experience providing technical support across a variety of IBM products and technologies, including communications, network, and systems management. For the last six years, he has worked for IBM ITSO, writing IBM Redbooks™ and creating and teaching workshops around the world on a variety of topics. In this position, he also works on various Tivoli customer projects as a Systems Architect and Consultant. Vasfi holds a master's degree in Engineering.

Frederic Marsaud is an IBM IT Pre-Sales Specialist working for the Tivoli Software Group in France for two years. He started working with IBM AIX® 5L™ for IBM Global Services, working four years as a Technical Engineer. He holds an AIX Certified Engineer Certification (June 2002). In 2004, he joined the Tivoli Software Group specializing in IBM Tivoli Management Framework, IBM Tivoli Monitoring, IBM Tivoli Enterprise Console®, and IBM Tivoli Monitoring for Transaction Performance. Currently, he works in promoting the IBM Tivoli Monitoring V6.1 and IBM Tivoli Workload Scheduler V8.3 products.

Haakan Gradin is a Technical Specialist with IBM Tivoli Software Group. He has more than 20 years of experience in systems management and has worked with Candle® and OMEGAMON® for more than 11 years. Over these years, Haakan has built an extensive expertise on the internal architecture of OMEGAMON products (both IBM z/OS® and distributed) and IBM Tivoli Monitoring solutions. Haakan is based in the U.K. working in the financial services sector.

John Willis is a Tivoli Enterprise Certified Instructor, Consultant, and Lead Architect for Gulf Breeze Software (`http://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 about IBM Tivoli Distributed Monitoring, IBM Tivoli Enterprise Console, and the workbench. His current area of expertise includes working with Common Information Model (CIM) and Windows® Management Instrumentation (WMI).

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

Arzu Gucer, Elizabeth Barnes
International Technical Support Organization, Austin Center

Caleb Marcinkovich, Elizabeth Purzer, Liza Poggemeyer, Divyesh Vaidya
IBM U.S.

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

Overview of IBM Tivoli Monitoring Express V6.1

This chapter introduces the concepts and components behind IBM Tivoli Monitoring Express V6.1. If you are new to IBM Tivoli Monitoring solutions, we recommend that you refer to the IBM Redbook titled *Deployment Guide Series: IBM Tivoli Monitoring 6.1*, SG24-7188, for more information about IBM Tivoli Monitoring V6.1.

This chapter includes the following topics:

- Challenges faced by mid-market companies
- IBM Tivoli Monitoring Express V6.1 solution
- Components of IBM Tivoli Monitoring Express V6.1
- Platform support matrix for IBM Tivoli Monitoring Express V6.1 (general availability)
- Differences between IBM Tivoli Monitoring V6.1 and IBM Tivoli Monitoring Express V6.1

1.1 Challenges faced by mid-market companies

Few mid-market companies exist today that are not dependent on computing systems to run their business. In many cases, these are deemed critical to the success of the business. The ability to effectively respond to computing resource needs and requirements, whether hardware or software, is paramount to the success of these companies.

The challenges faced by small and medium businesses (SMB) include:

- Limited people resources to manage the growing complexity of a company's heterogeneous environment
- Limited buying power, often due to low IT budget
- Limited IT capability in terms of staffing and skills
- Lack of affordable solutions
- Lack of vision and strategic direction, which in turn increases costs
- Revenue loss when the systems that business applications depend on are not readily available

To complicate matters, the problems that arise are often not understood completely because they are usually not caused by a single event, but rather a series of events or multiple concurrent events that might not be clear when the problem occurs. Usually, without a clear view of the computing infrastructure and software services as a whole, only the symptoms are addressed, and not the root cause. This leads to redundant, expensive, and inefficient efforts.

In order to monitor and react to any event that can affect the delivery of critical business services in a mid-market company, you should focus on how to identify the common resources that must be monitored, how they should be monitored, and the common corrective actions that can be used to respond to problematic situations. In the process of understanding these monitoring requirements, the solution described here shows how to build a monitoring console that can be easily used by support personnel.

Defining the business needs of SMBs

Monitoring computing resources is an essential task for all businesses. This section defines the key driving forces behind an effective SMB enterprise.

Take into account the following business needs of SMBs when planning and defining SMB enterprise management:

- Need to increase revenues, reduce costs, and compete more effectively

  The investment in IT systems should constitute a significant percentage of corporate expenses.

- Value and return on investment (ROI)

  There is a need for rapid ROI in as little as 60 days with solutions that enable up-selling and cross-selling of products and high-value services.

- Affordable solution

  SMBs should get attractive prices, with no additional charge for additional monitoring agents.

- Ability to tailor to the size of the company

  There is a need for an express solution that is transparent to the business solutions being delivered, besides being proactive in detecting, resolving, and escalating potential issues that might impact the business service being delivered.

- Ease of integration and installation

  We built an express solution that enables the IT support personnel to view and understand their company's IT infrastructure that consists of common servers and services for e-mail, directory, and Web sites running in a heterogeneous operating system (OS) environment.

- Easily manageable

  An intuitive user interface, simple administration, and out-of-the box reporting help simplify the ongoing monitoring and management processes. Self-managing solutions that involve less system intervention, less disruption to the business, and provide greater IT efficiency for sustained business growth help save time and money.

1.2 IBM Tivoli Monitoring Express V6.1 solution

IBM Tivoli Monitoring Express V6.1 manages the performance and availability of small-to-medium business systems and application resources. It helps the IT infrastructure of these businesses move at the same speed as their business expectations. IBM Tivoli Monitoring Express V6.1 provides reports that can be used to track trends and troubleshoot problems. In addition to this, achieving faster time-to-value becomes possible through simple installation and easy-to-use interfaces.

You can customize this solution to accommodate different requirements. This includes applying standards for monitoring thresholds and recovery processes.

The IBM Tivoli Monitoring Express V6.1 solution has the following advantages:

- Simple, one-click installation

  A Launchpad facilitates simple and quick installation.

- Ease of deployment

  The remote configuration of management agents streamlines deployment.

- Simple, end-to-end management of heterogeneous environments from a single, consistent interface

  Centralized and simplified management facilitates quick identification and resolution of IT resources, applications, and network problems, thereby increasing the availability and reducing the costs associated with keeping the business up and running.

- Powerful visualization

  Proactive identification of trends helps avoid outages and performance issues.

- Expert advice and automated actions to speed resolution

  Expert advice provides a detailed description of problems and the recommended recovery actions, in addition to enabling IT operations to resolve known errors and freeing up subject matter experts to work on other value-add activities.

- Built-in real-time and historical reporting, that is, reports creation included at no charge

  The ability to view real-time and historical data side by side enables you to identify changes in performance. All data, whether real-time, short-term operational, long-term operational, or aggregated, is available from a unique console, that is, Tivoli Enterprise Portal.

- Complete correlation

  Out-of-the-box situations provided by IBM Tivoli Monitoring Express V6.1 allow for correlation of metrics that help avoid event overload and false alerts.

- Predefined situations

  You can take immediate advantage of these situations, which include preset thresholds, sampling intervals, Boolean logic, and expert advice.

This solution is designed and priced to specifically meet the needs of mid-market businesses. The designing and pricing features include:

- Single packaged offering, with IBM DB2® Express included at no additional charge.
- Single server license.
- Single management server deployment.
- No charge for additional IBM Tivoli Monitoring Express V6.1 management agents.
- Support of up to 100 physical servers, bounded by 4-processor per server machines. The 4-processor per server is the maximum limit. There is, however, no limit on IBM Tivoli Monitoring Express V6.1 agents.

When you obtain an IBM Tivoli Monitoring Express V6.1 license after running a 60-day trial version, you can download additional agent products that are not included on the IBM Tivoli Monitoring Express V6.1 CDs:

- IBM Tivoli Monitoring for Databases V6.1 includes monitoring agents for DB2, Oracle, Microsoft® SQL Server, and Sybase.
- IBM Tivoli Monitoring for Messaging and Collaboration V6.1 includes a monitoring agent for Microsoft Exchange Server.
- IBM Tivoli OMEGAMON XE for Microsoft .NET includes monitoring agents for BizTalk® Server, Commerce Server, Content Management Server, .NET Framework, Host Integration Server, Internet Security and Acceleration Server 2000, Internet Security and Acceleration Server 2004, SharePoint® Portal Server, and Universal Description, Discovery, and Integration (UDDI) services.

Important: You can download additional IBM Tivoli Monitoring Express V6.1 agents (such as virtual servers, Citrix, Microsoft Cluster Manager Agents) through the license Web site. Licensing of SAP and Siebel monitoring agents was under discussion at the time of writing this redbook. Contact your IBM representative for the URL of the licensing Web site and if you want to use SAP and Siebel monitoring agents.

Additional service opportunities are available through the IBM Tivoli Open Process Automation Library for Business Partners, available on the Web at:

http://www-18.lotus.com/wps/portal/tm

1.3 Components of IBM Tivoli Monitoring Express V6.1

A set of modules built on top of IBM Tivoli Monitoring Express V6.1 provide a comprehensive set of solutions for companies facing the challenge of monitoring composite application infrastructures.

This section introduces the various components that provide the technology for IBM Tivoli Monitoring Express V6.1.

1.3.1 Tivoli Monitoring Services

Tivoli Monitoring Services is the framework for IBM Tivoli Monitoring Express V6.1. This section provides details about all the components and describes how they interact. The components include, but are not limited to, the following solutions:

- Tivoli Enterprise Monitoring Agent
- Tivoli Enterprise Monitoring Server
- Tivoli Enterprise Portal Server
- Tivoli Enterprise Portal

Figure 1-1 shows the various components of IBM Tivoli Monitoring Express V6.1.

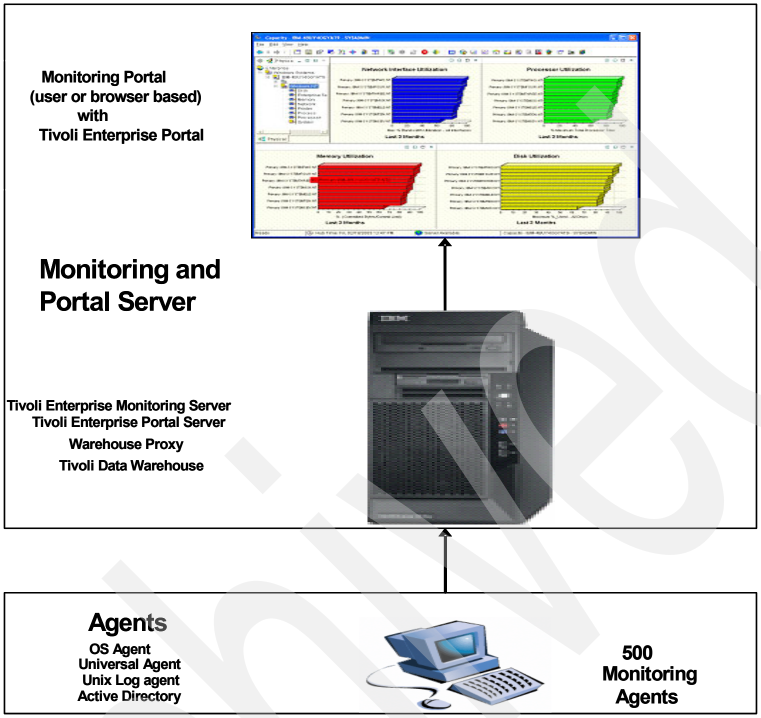


Figure 1-1 IBM Tivoli Monitoring Express V6.1 components

Tivoli Enterprise Monitoring Agent

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

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

IBM Tivoli Monitoring Express V6.1 includes a set of agents that perform the monitoring and data collection functions of the product. These are:

- Monitoring Agent for Windows OS
- Monitoring Agent for Linux® OS (Intel®)
- Monitoring Agent for UNIX® Logs

- Monitoring Agent for i5/OS®

  The Monitoring Agent for i5/OS is not included in the agent depot for remote deployment from the server. It must be installed locally on an IBM i5/OS system. Refer to *IBM Tivoli Monitoring V6.1 i/5 OS Agent User's Guide,* SC32-9448, for information about installing, configuring, and using the i5/OS Agent.

- Monitoring Agent for UNIX OS
- Monitoring Agent for Active Directory
- Tivoli Universal Agent for Windows, Linux Intel, and UNIX

  You can configure this monitoring agent to monitor any data you collect. It enables you to integrate data from virtually any platform and source, such as custom applications, databases, systems, and subsystems.

- Warehouse Proxy agent for short-term data reporting

  This is a unique agent that performs only one task, that is, collecting and consolidating all historical data collections from the individual agents to store in Tivoli Data Warehouse.

  Restriction: IBM Tivoli Monitoring Express currently supports only the Warehouse Proxy agent under the Microsoft Windows platform.

- Summarization and Pruning agent for historical long-term data reporting

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

  Restriction: IBM Tivoli Monitoring Express V6.1 currently supports only the Summarization and Pruning agent under the Windows platform.

At the time of general availability, the following OS agents will be supported:

- IBM AIX 5L Versions 5.1, 5.2, 5.3
- Sun™ Solaris™ 8, 9, 10
- HP UX 11i
- Microsoft Windows 2000 Server, Advanced Server
- Microsoft Windows XP
- Microsoft Windows Server 2003 32-bit (Standard Edition and Enterprise Edition) with Service Pack 1
- IBM OS/400® Versions 5.2, 5.3
- Red Hat Enterprise Linux (RHEL) 2.1

- Red Hat Enterprise Linux (RHEL) 4 (Intel)
- SUSE Linux Enterprise Server (SLES) 8 (Intel)
- SUSE Linux Enterprise Server (SLES) 9 (Intel)

Tivoli Enterprise Monitoring Server

Tivoli Enterprise Monitoring Server, referred to as monitoring server, acts as a collection and control point for alerts received from the agents and collects their performance and availability data.

At the time of general availability, the following platforms will be supported:

- Microsoft Windows 2000 Server, Advanced Server
- Microsoft Windows Server 2003 32-bit (Standard Edition and Enterprise Edition) with Service Pack 1

Tivoli Enterprise Portal Server

Tivoli Enterprise Portal Server, which is referred to as portal server and is placed between the client and the monitoring server, enables retrieval, manipulation, and analysis of data from the agents. It uses a DB2 or Microsoft SQL database to store data.

At the time of general availability, the following platforms will be supported:

- Microsoft Windows 2000 Server, Advanced Server
- Microsoft Windows Server 2003 32-bit (Standard Edition and Enterprise Edition) with Service Pack 1

Tivoli Enterprise Portal

The Tivoli Enterprise Portal client displays information through the use of workspaces in the form of charts and tables. With predefined workspaces, you can start monitoring activity and system status immediately. With a few clicks, 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, as is appropriate for your business.

You can launch Tivoli Enterprise Portal from a Microsoft Internet Explorer browser or install it as a client application on a workstation.

Assuming a default installation, use the following URL for the browser mode Tivoli Enterprise Portal client:

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

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

Important: IBM Tivoli Monitoring Express V6.1 supports only Internet Explorer on the Windows platform in browser mode.

At the time of general availability, the following platforms will be supported:

- Microsoft Windows 2000 Professional
- Microsoft Windows 2000 Server, Advanced Server
- Microsoft Windows XP
- Microsoft Windows Server 2003 32-bit (Standard Edition and Enterprise Edition) with Service Pack 1

1.3.2 IBM Tivoli Data Warehouse V2.1

Tivoli Data Warehouse V2.1 is the database storage that contains all the historical data collection. The data warehouse is located on an IBM DB2 or Microsoft SQL database.

For more information about Tivoli Data Warehouse V2.1, refer to Chapter 3, "Historical summarized data" on page 101.

1.4 Platform support matrix for IBM Tivoli Monitoring Express V6.1 (general availability)

Table 1-1 displays the full support matrix for the general availability code release of IBM Tivoli Monitoring Express V6.1. It provides an overview of platform support and planned support direction.

Important: Always review the latest *Release Notes* packaged with IBM Tivoli Monitoring Express V6.1 for the latest platform support information.

Table 1-1 Platform support matrix for IBM Tivoli Monitoring Express V6.1

Supported OS	TEMA[a]	TEMS[b]	TEP[c]	TEPS[d]	Warehouse Proxy	S&P agent[e]
AIX 5L V5.1 (32/64 bit)	X					
AIX 5L V5.2 (32/64 bit)	X					
AIX 5L V5.3 (32/64 bit)	X					
Solaris 8 (32/64 bit)	X					

Supported OS	TEMA[a]	TEMS[b]	TEP[c]	TEPS[d]	Warehouse Proxy	S&P agent[e]
Solaris 9 (32/64 bit)	X					
Solaris 10 (32/64 bit)	X					
HP-UX 11i (32/64 bit)	X					
Windows 2000 Professional (32 bit)	X		X			X
Windows 2000 Server (32 bit)	X	X	X	X	X	X
Windows 2000 Advanced Server (32 bit)	X	X	X	X	X	X
Windows XP (32 bit)	X		X		X	X
Windows Server 2003 Enterprise Edition with SP 1 (32 bit)	X	X	X	X	X	X
Windows Server 2003 Standard Edition with SP 1 (32 bit)	X	X	X	X	X	X
OS/400 5.2	X					
OS/400 5.3	X					
RHEL 2.1 AS/ES Intel (32 bit)	X					
SLES 8 Intel (32 bit)	X					
SLES 9 Intel (32 bit)	X					
RHEL 2.1 WS Intel	X					

a. Tivoli Enterprise Monitoring Agent
b. Tivoli Enterprise Monitoring Server
c. Tivoli Enterprise Portal
d. Tivoli Enterprise Portal Server
e. Summarization and Pruning agent

Note: Windows XP is supported as a demonstration platform in IBM Tivoli Monitoring V6.1, but not in IBM Tivoli Monitoring Express V6.1.

1.5 Differences between IBM Tivoli Monitoring V6.1 and IBM Tivoli Monitoring Express V6.1

IBM Tivoli Monitoring Express V6.1 is a compact, yet functional, version of IBM Tivoli Monitoring V6.1.

IBM Tivoli Monitoring Express V6.1 solution provides a solid foundation for the development of management solutions addressing the complex needs of today's IT infrastructures.

The following features distinguish IBM Tivoli Monitoring Express V6.1 from IBM Tivoli Monitoring V6.1:

- In IBM Tivoli Monitoring Express V6.1, the management infrastructure is based on only Windows. Linux will be supported in the next release.
- The heterogeneous environment of IBM Tivoli Monitoring Express V6.1 includes Windows, UNIX, Linux, i5/OS. There is zSeries® support.
- In IBM Tivoli Monitoring Express V6.1, the Policy editor has been removed, as opposed to IBM Tivoli Monitoring V6.1. In IBM Tivoli Monitoring V6.1, many monitoring agents provide predefined policies that you can use or modify to suit your environment.
- In IBM Tivoli Monitoring Express V6.1, there is no remote management server, only a single management server deployment. It can scale up to 100 servers (initial release handled through licensing).
- IBM Tivoli Monitoring Express V6.1 will be integrated only with other Tivoli Express offerings. There is no integration with IBM Tivoli Enterprise Console, IBM Tivoli Business System Manager, IBM Tivoli Service Level Advisor, and so on.
- The Tivoli Data Warehouse middleware supports only DB2 Express and Microsoft SQL.

Note: IBM Tivoli Monitoring Express V6.1 can interface network products such as IBM Tivoli NetView® through Simple Network Management Protocol (SNMP) data provider through IBM Tivoli Universal Agent.

2

Product architecture and deployment best practices

This chapter explains the architecture of IBM Tivoli Monitoring Express V6.1 and how each component operates within an IBM Tivoli Monitoring Express V6.1 installation. It also explores an architectural design of the IBM Tivoli Monitoring Express V6.1 solution. The overview section covers IBM Tivoli Monitoring Express V6.1 agent deployment using several unique strategies.

In this chapter, we discuss the following topics:

- Implementation scenarios of IBM Tivoli Monitoring Express V6.1
- Scalability of IBM Tivoli Monitoring Express V6.1
- Installing Tivoli Enterprise Management Agents
- Installing IBM Tivoli Monitoring Express V6.1 and related components
- Configuring IBM Tivoli Monitoring Express V6.1 components
- Installing IBM Tivoli Monitoring Express V6.1 agents
- Uninstalling IBM Tivoli Monitoring Express V6.1

2.1 Implementation scenarios of IBM Tivoli Monitoring Express V6.1

This section provides a realistic understanding of architecture design. It also helps you set up an instance of an IBM Tivoli Monitoring Express V6.1 solution.

The IBM Tivoli Monitoring Express V6.1 installation is a fundamental design, using only the minimum required components. The out-of-box monitoring collections, graphical user interface (GUI) presentation layer, historical data collection, and robustness provide a complete monitoring solution with a modest total cost of ownership (TCO).

You can implement the IBM Tivoli Monitoring Express V6.1 solution with minimum hardware requirements. The IBM Tivoli Monitoring Express V6.1 installation media includes a number of monitoring agents that you can immediately add to your environment when you install the product.

When you install the IBM Tivoli Monitoring Express V6.1 server components using the IBM Tivoli Monitoring Express Launchpad, the following components are deployed on a single machine:

- Tivoli Enterprise Monitoring Server: The first component installed to begin building the IBM Tivoli Monitoring Services foundation.
- Tivoli Enterprise Portal Server: A repository for all graphical presentation of monitoring data.
- Tivoli Enterprise Portal: A Java™-based user interface component of the presentation layer that connects to the Tivoli Enterprise Portal Server to view all monitoring data collections. The two modes of operation it offers are a Java desktop client and an Hypertext Transfer Protocol (HTTP) browser.
- Tivoli Data Warehouse: The database storage containing all the historical data collection.
- Tivoli Warehouse Proxy agent: Collects and consolidates all the historical data collections from the individual agents to store in the Tivoli Data Warehouse.
- Tivoli Warehouse Summarization and Pruning agent: A unique agent that performs the aggregation and pruning functions for the historical raw data on Tivoli Data Warehouse.
- Tivoli Enterprise Monitoring Agents: Data collectors within your monitoring express solution. They are installed to gather data from one or more systems you need to monitor. The default monitoring agents installed are:
  - Operating system agent

– Active Directory agent

The Active Directory agent monitors the Active Directory component of the Microsoft Windows OS.

> **Important:** While implementing the Tivoli Active Directory agent, register the iadstools.dll file on the target machine. Use **`regsvr32 iadstools.dll`** in a MS-DOS® command. For more information about the Tivoli Active Directory agent requirements, refer to Chapter 2 of the *IBM Tivoli Monitoring for Active Directory User's Guide*, SC32-9444.

– Tivoli Universal Agent

IBM Tivoli Universal Agent is a generic agent used to collect data and monitoring systems and applications in your network. It can interface with a variety of different sources to monitoring databases, log files, SNMP traps, and so on. It provides multiple data providers such as HTTP, file, Open Database Connectivity (ODBC), post, script, SNMP, socket, and API for maximum flexibility on virtually any platform.

The Universal Agent can also interface on an API level with another application. This makes it very powerful. If used properly, it can manage almost any kind of monitoring scenario.

IBM Tivoli Universal Agent enables you to carry out the monitoring of remote systems without deploying local monitoring agents.

For more information about Tivoli Universal Agent, see 5.12.1, "What is Tivoli Universal Agent?" on page 295.

Figure 2-1 on page 16 depicts the topology design of IBM Tivoli Monitoring Express V6.1. It provides an overview of each component connected to IBM Tivoli Monitoring Express V6.1 and depicts the following services:

- The monitoring and portal server, through the Tivoli Enterprise Monitoring Server, acts as a collection and control point for alerts received from Tivoli Enterprise Monitoring Agent and collects their performance and availability data. The server is responsible for tracking the heartbeat status for all agents connected to it. An agent is a lightweight application that is deployed on servers to be monitored.
- The monitoring and portal server, through the Tivoli Enterprise Portal Server, provides a customizable, graphical view of the monitored infrastructure. It consists of all the user IDs and user access controls for the monitoring workspaces and provides the core presentation layer that allows for retrieval, manipulation, analysis, and preformatting of data.

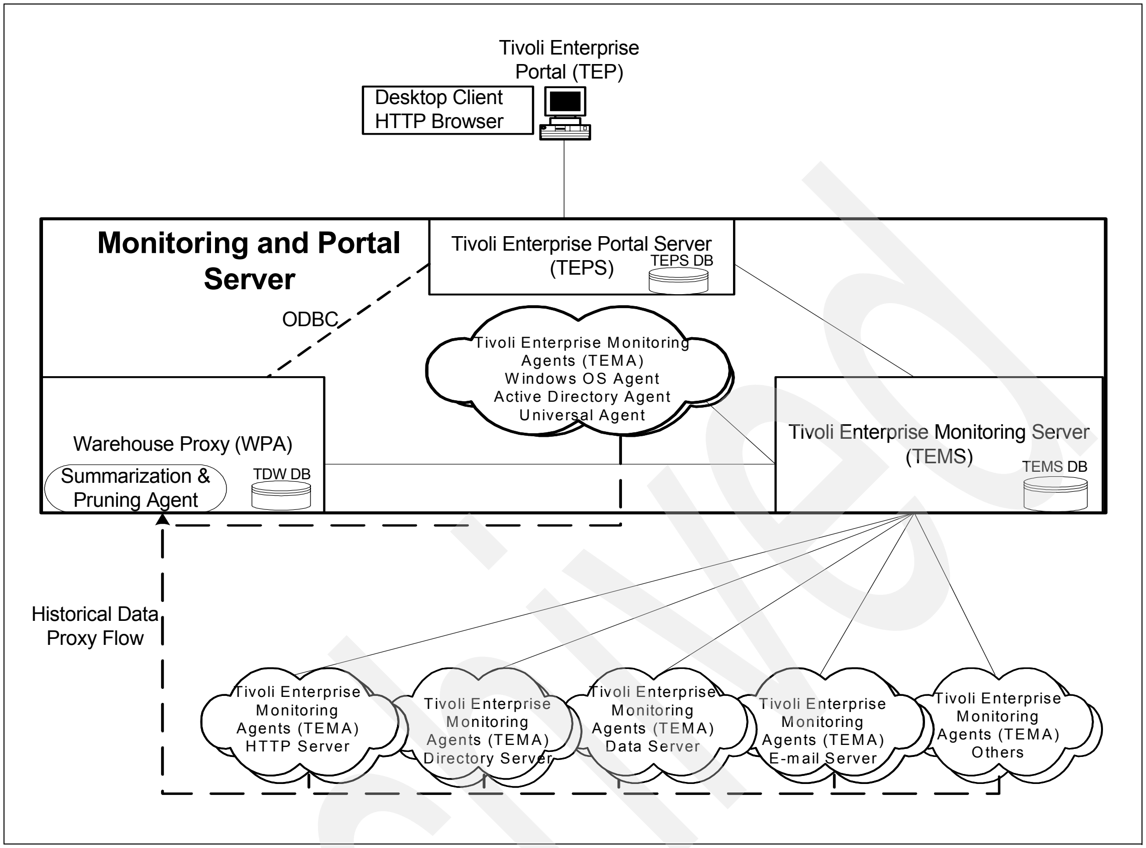


Figure 2-1 IBM Tivoli Monitoring Express V6.1 topology design

The IBM Tivoli Monitoring Express V6.1 installation supports approximately 500 managed systems, that is, agents. A managed system is the component for which a Tivoli Enterprise Monitoring Agent is installed.

The Tivoli Enterprise Monitoring Agents are installed on the system or subsystem requiring data collection and monitoring. The agents are responsible for gathering data and distributing attributes to the monitoring servers, including initiating the heartbeat status.

> **Note:** All the managed systems can contain an OS agent and non-OS agents.

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 Express V6.1 installation. All the agents connect directly to the Tivoli Monitoring server through the component Tivoli Enterprise Monitoring Server.

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 that processing rate is on target.

IBM Tivoli Monitoring Express V6.1 is ideal for small installations. Immediately after installation, it begins to leverage the best practice functionality. Default situations start running, and if historical data collection is turned on, the default attribute groups begin analysis and warehousing.

For a good solution overview, install IBM Tivoli Monitoring Express V6.1 on a single machine running Microsoft Windows 2000 or Windows 2003 Service Pack 1.

For demonstration purposes, we describe the IBM Tivoli Monitoring Express V6.1 environment with two types of database software:

- DB2 Express server on Windows Server 2003 with all security updates applied
- Microsoft SQL Server on Windows Server 2003 with all security updates applied

Figure 2-2 depicts the interconnections between the various components at their simplest.

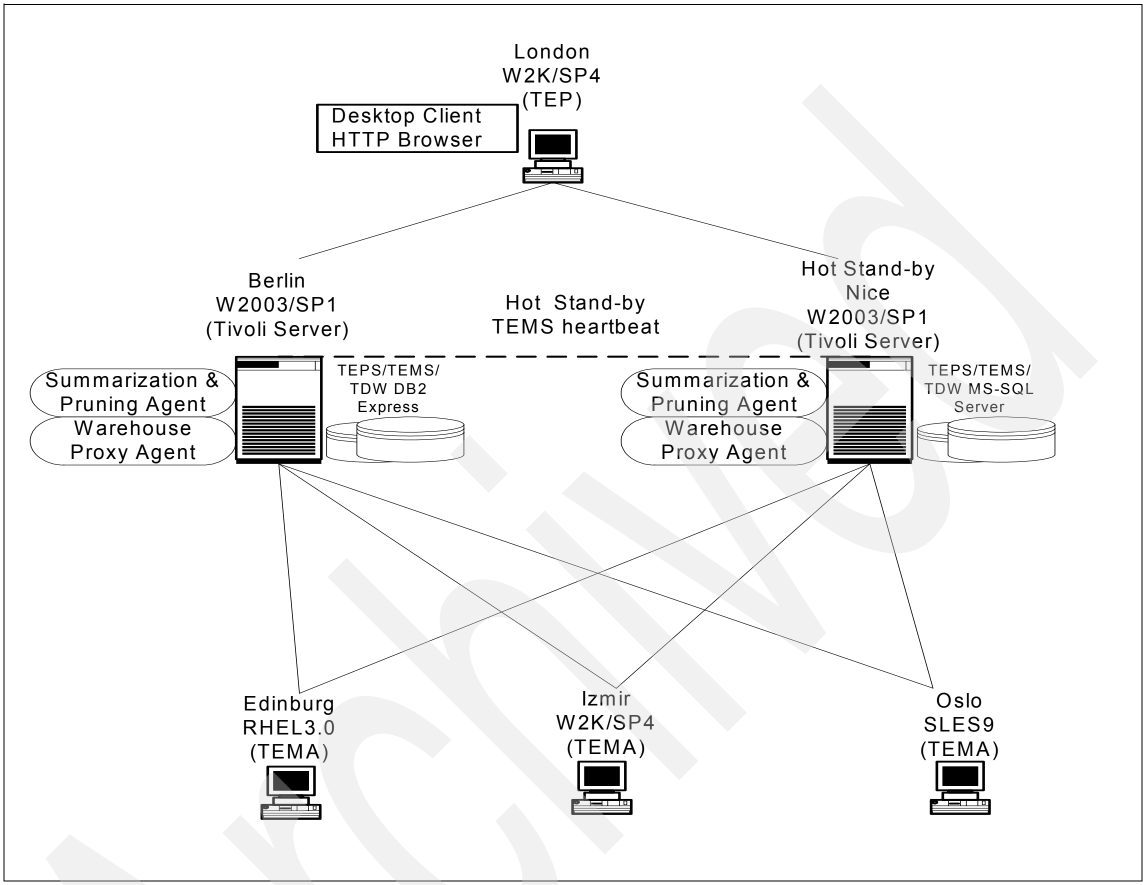


Figure 2-2 IBM Tivoli Monitoring Express V6.1 lab topology

To cover the various topics discussed throughout this book, we implement an IBM Tivoli Monitoring Express V6.1 installation with a topology design that incorporates all related content. This architecture covers all the components that make up an IBM Tivoli Monitoring Express V6.1 installation. To ensure the accuracy of the implementation and best practices, the environment contains a proportionate selection of heterogeneous hardware configurations with varying types of operating system platforms and levels.

Table 2-1 shows the hardware and software configuration of our lab environment.

Table 2-1 Lab hardware and software configuration

Server	OS	CPU	Memory	Hard disk	Main components
berlin	Windows 2003 SP1	Pentium® 4 3 GHz	2 GB	32 GB	Tivoli Enterprise Portal Server, Tivoli Enterprise Monitoring Server, Tivoli Data Warehouse, Tivoli Enterprise Monitoring Agent
nice	Windows 2003 SP1	Pentium 4 3 GHz	1.5 GB	32 GB	Tivoli Enterprise Portal Server, Tivoli Enterprise Monitoring Server, Tivoli Data Warehouse, Tivoli Enterprise Monitoring Agent
edinburg	RHEL 3U1	Pentium 4 1.8 GHz	1 GB	40 GB	Tivoli Enterprise Monitoring Agent
oslo	SLES 9	Pentium 4 1.8 GHz	1 GB	40 GB	Tivoli Enterprise Monitoring Agent
izmir	Windows 2003 SP1	Pentium 4 1.8 GHz	1 GB	22 GB	Tivoli Enterprise Monitoring Agent

You have the option to install a monitoring and portal server as a hot standby node, as shown in Figure 2-2 on page 18. This is recommended, but not required for the SMB installation, especially if cost restrictions exist for hardware deployment. Always consider the hot standby monitoring and portal server because it offers failure protection with a minimum increase in total cost of ownership. Implementing such an architecture in the early stages allows for growth and scalability. Furthermore, this design builds around the IBM Tivoli Monitoring Express V6.1 built-in fail-over capabilities.

What is the hot standby process in IBM Tivoli Monitoring Express V6.1 solution?

Overall, running the monitoring and portal server as a hot standby node (referred to as a hot standby Tivoli Enterprise Monitoring Server) means that you are running a second hub Tivoli Enterprise Monitoring Server (TEMS2) in parallel to the current hub (TEMS1). The hot standby node always must be configured as a hot standby hub Tivoli Enterprise Manager Server. Agents must be configured first to be able to connect to two hub Tivoli Enterprise Monitoring Servers.

Suppose that the Tivoli Enterprise Monitoring Agents are already running, before the hub Tivoli Enterprise Monitoring Servers are started. At their initialization, the monitoring agents and Tivoli Enterprise Monitoring Server will try to connect to the Tivoli Enterprise Monitoring Server that is specified first in their configuration. If the connection fails, at next interval, the monitoring agent will try to connect to the other hub Tivoli Enterprise Monitoring Server, and so on.

The first hub Tivoli Enterprise Monitoring Server (TEMS1) to start automatically becomes the primary hub in the configuration. Do not start the secondary hub within 10 minutes after the primary started; otherwise, agents might first try to connect to the secondary hub. (Agents 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 monitoring agent. When monitoring agents 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 the 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 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 monitoring agents that make the switch to the new primary hub. TEMS2 also restarts situations at this time. All sampled situations are reevaluated and raised if needed. Pure events might not be raised if there is no longer a source for the alert.

Agents will detect the “hub down” condition when making their heartbeat connection to the hub. At that time, they try to connect to the secondary Tivoli Enterprise Monitoring Server that was defined during their configuration. If the agent 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.

Tivoli Enterprise Monitoring Server hot standby is in no way an immediate (in microseconds) or user-transparent solution; a typical fail-over scenario takes from a few minutes to 30 minutes to complete, depending on the options taken during the implementation. As usual, the faster you want the failover to complete, the more it will cost in terms of system resources.

At fail-over time, end users see a pop-up message on the Tivoli Enterprise Portal signalling that the Tivoli Enterprise Portal Server has lost contact with the hub Tivoli Enterprise Monitoring Server. This is because the Tivoli Enterprise Portal Server currently does not yet support Tivoli Enterprise Monitoring Server failover. Configuration changes can be taken to alleviate the impact to end users, but at the very least, they will be required to log on again.

Note: The SMB installation supports approximately 500 managed systems (referred to as agents). 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 Express V6.1 installation.

2.2 Scalability of IBM Tivoli Monitoring Express V6.1

A distributed networking infrastructure inherits scalable characteristics by design. After all, a distributed system is built to expand and shrink through the increases and decreases in hardware capacity. Scalability is not the same as performance tuning. Performance tuning deals with increasing output from the 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 should, in theory, extend to infinity. However, as the distributed systems increase in scalability, performance loss might also increase to an unsustainable boundary. IBM Tivoli Monitoring Express V6.1 follows the basic scalable characteristic in this design.

The hardware and software configuration should scale to handle the monitoring of up to 500 monitoring agents.

Information about estimating the required size of your Tivoli Data Warehouse database is available in the IBM Redbook, *Getting Started with IBM Tivoli Monitoring 6.1 on Distributed Environments*, SG24-7143.

For IBM Tivoli Monitoring Express V6.1, 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 Express V6.1 and strategically working through them facilitates obtainable goals.

From a scalability standpoint, the monitoring and portal server plays the key role. As the architect of an IBM Tivoli Monitoring Express V6.1 implementation, consider the following factors:

- The number of physical hosts and platform types included

- The number and type of applications and OS per host
- The geographical topology of the environment, particularly in relation to where the managed systems will reside
- The estimated number of events generated, thresholds that will be deployed, or both
- The estimated number of Tivoli Enterprise Portal users and the expected type of usage, that is, heavy reporting, frequent real time updates, and so on

Combine the information generated from these points with the scalability guidelines established for the initial release of IBM Tivoli Monitoring.

For additional support and information, refer to the following resources:

- The IBM Solutions Consultant Express Tool simplifies the solution design experience by codifying IBM Patterns for e-business into a wizard-based tool that guides the user through an analysis of the business requirements and customer's IT environment. Based on the user's answers, the tool then recommends the appropriate patterns and associated Express products that best fit the problem space. See (in the Toolbox area):

  http://www.ibm.com/partnerworld/solutionsbuilder

- The Technology Assessment Tool helps you help your clients identify their IT maturity in terms of On Demand Operating Environment capabilities. It lets users better understand where they are today and determine where they want to go on their journey to becoming an On Demand Business. The tool provides a technology blueprint that is based on the user's current capabilities and IT goals. See (in the Toolbox area):

  http://www.ibm.com/partnerworld/solutionsbuilder

- The Virtual Innovation Center (VIC) provides training, sales materials, and support services for IBM Tivoli Monitoring in addition to other IBM Software products at no charge to you or your company. See:

  http://www.ibm.com/partnerworld/vic

The following scalability metrics are from the verification testing performed on IBM Tivoli Monitoring Express V6.1 (GA). These numbers represent the actual test synopsis validation and 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 Express V6.1installations are unique and require surveillance during deployment.

Table 2-2 on page 23 classifies the extensive metrics for IBM Tivoli Monitoring Express V6.1. These metrics measure the apex for the IBM Tivoli Monitoring Express V6.1 components with respect to load quantity. Each metric represents one installation instance.

Table 2-2 Extensive metrics

IBM Tivoli Monitoring Express V6.1 components	Verified metric
Managed systems per monitoring and portal server	500
Agents storing historical data at monitoring and portal server	250
Consoles per monitoring and portal server	50
Total situations	1,500 (30/agent)

Important: These metric values do not represent the actual hard limits in IBM Tivoli Monitoring Express V6.1. These numbers are derived from what was actually tested, and 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 3, “Historical summarized data” on page 101.

2.3 Installing Tivoli Enterprise Management Agents

Several techniques exist for installing the Tivoli Enterprise Management Agents. This section summarizes three common practices that you can employ to install managed systems with an installation.

Keep the following factors in mind while installing the agents:

- The total number of physical systems and the total number of agents deployed to each of the systems
- The network bandwidth and latency between the monitoring and portal server and the monitoring agent
- The size of the IBM Tivoli Monitoring installation
- The connectivity to the managed systems

Built-in deployment controller of Tivoli Monitoring Express V6.1

IBM Tivoli Monitoring Express V6.1 offers an easy and efficient deployment mechanism to push OS agents and non-OS agents to remote systems. This mechanism also offers the ability to upgrade the agent, in addition to providing a powerful built-in tool for intelligent agent upgrades through the GUI or command line.

Figure 2-3 shows the architecture of the IBM Tivoli Monitoring Agent components. The functionality of the agent components is divided among the Tivoli Enterprise Portal Server, Tivoli Enterprise Monitoring Server, and the OS agent, respectively.

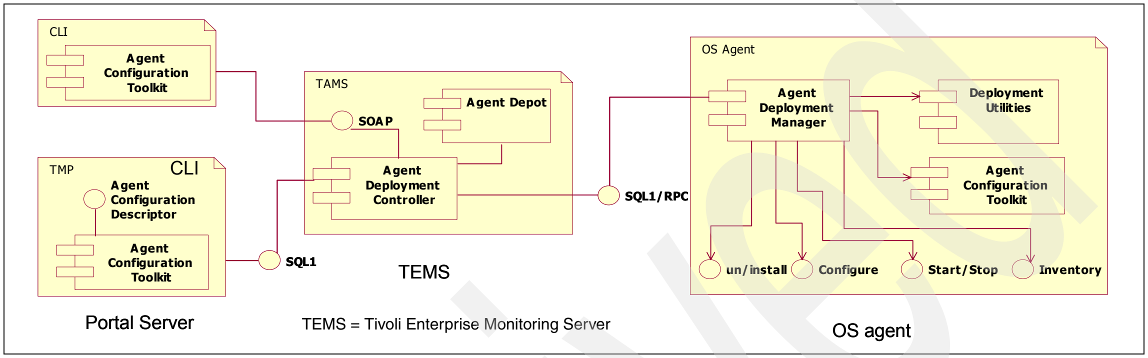


Figure 2-3 Agent deployment architecture

IBM Tivoli Monitoring OS agents, implemented as a dynamic link library (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 resides on the monitoring and portal server. Before you deploy any agents from a monitoring server, populate the agent depot with bundles. A bundle is the agent installation image and prerequisites, if any. Load the agents into the agent depot at the time of installation.

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. Customize the agent depot based on the types of bundles you want to deploy and manage from a given 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 the agent bundles using remote procedure calls (RPCs). All the 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.

Important: We recommend that you consider the following points:

- To allow remote deployment, the target system must support, and be configured for, at least one of the following protocols:

  - Server Message Block (SMB)
  - Secure Shell Protocol (SSH)
  - Remote Execution Protocol (REXEC)
  - Remote Shell (RSH)

  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 another program located in another computer in a network, without having to understand the network details. A procedure call is also known as a function call or a subroutine call.
- SQL1 is the SQL implementation based on the ANSI-1989 SQL1 standard.

You can target the deployment controller commands to a specific system or a managed system list. The deployment controller manages the interaction with the management agent (OS agent). It manages the receiving and aggregating of results from multiple targets and forwards the requests to the monitoring and portal server through the Tivoli Enterprise Monitoring Server component, in addition to queuing up the requests for scalability. You can initiate the installation, uninstallation, and upgrade processes.

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 the configuration data and provides a set of utilities that enable agent deployment to configure the agents.

The Agent Configuration Toolkit and the deployment controller communicate through SOAP.

A program running on one kind of OS, such as Windows 2000, can communicate with a program in the same or another type of OS, such as Linux, with the help of SOAP by using the World Wide Web 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 systems, HTTP and XML provide a ready solution to the problem of how programs running under different OSs in a network can communicate with each other.

SOAP specifies 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 on information. It also specifies how the called program can respond.

The SOAP advantage 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. Because HTTP requests are usually allowed through firewalls, programs using SOAP to communicate can be sure of communicating with programs anywhere.

2.4 Installing IBM Tivoli Monitoring Express V6.1 and related components

This section describes how to install IBM Tivoli Monitoring Express V6.1 and related components. For information about the supported databases and software requirements, refer to *Getting Started with IBM Tivoli Monitoring Express,* SC32-1903. Table 2-3 lists the product codes of IBM Tivoli Monitoring Express V6.1.

Table 2-3 Complete product code listing of IBM Tivoli Monitoring Express V6.1

Component	**Product code**	**Seed**
Active Directory monitoring agent	3z	Yes
Windows OS monitoring agent	nt	Yes
Linux OS monitoring agent	lz	Yes
Universal Agent	um	Yes
IBM DB2 monitoring agent	ud	Yes
IBM i5/OS monitoring agent	a4	Yes
Microsoft Exchange Server monitoring agent	ex	Yes
Microsoft SQL Server monitoring agent	oq	Yes
Sybase Server monitoring agent	oy	Yes
UNIX OS monitoring agent	ux	Yes
Warehouse Proxy	hd	
Warehouse Summarization and Pruning agent	sy	
Oracle monitoring agent	or	Yes

Table 2-4 shows the suggested software and hardware requirements to handle the monitoring of 300 to 500 monitoring agents.

Table 2-4 Suggested software and hardware

Server type	Software components	Software prerequisites	Suggested hardware
Monitoring and portal server	IBM Tivoli Monitoring Express V6.1 [a]	Microsoft Windows 2003 Server	Intel Pentium 4 3 GHz 3 GB memory 1 GB free disk space **Note:** This sizing recommendation does not include capacity for the warehousing features of IBM Tivoli Monitoring.
E-mail and directory server (already in the environment, but will have additional software added to it for this solution)	IBM Tivoli Monitoring Express V6.1 (Windows OS, Exchange, and Active Directory agents only)	Microsoft Windows Server 2003 Microsoft Active Directory and Microsoft Exchange Server 2003	Server already exists in customer environment.[b]
Database server (already in the environment, but will have additional software added to it for this solution)	IBM Tivoli Monitoring Express V6.1 (Windows OS, DB2, and Microsoft SQL agents only)	Microsoft Windows Server 2003 IBM DB2 Universal Database™ Enterprise Server Edition, V8.2 Microsoft SQL Server	Server already exists in customer environment.[b]
Web server (already in the environment, but will have additional software added to it for this solution)	IBM Tivoli Monitoring V6.1 (Linux OS and Universal Agent only)	SUSE Linux Enterprise Server V9 IBM HTTP Server V6.1 Web server plug-in for IBM WebSphere® Application Server V6.0	Server already exists in customer environment.[b]
Web application server (already in the environment, but will have additional software added to it for this solution)	IBM Tivoli Monitoring V6.1 (Windows OS and Universal Agent only)	Microsoft Windows Server 2003 IBM WebSphere Application Server Express V6.0	Server already exists in customer environment.[b]

a. IBM Tivoli Monitoring Express V6.1 is also supported on the following platforms: IBM AIX 5L, Sun Solaris, HP-UX, Red Hat Linux, SUSE Linux, and Linux on zSeries.
b. The Tivoli monitoring agents (lightweight applications with small memory footprints) installed on the existing customer servers should not require significant additional resources.

Special considerations

You must decide how the monitoring servers are to be named. In general, create names that are short but meaningful within your environment. Use the following guidelines:

- Each name must be unique. One name cannot match another monitoring server name for its entire length. (For example, ibm and ibmremote are unique and permitted.)
- Each name must begin with an alpha character. No blanks or special characters ($#@) can be used.
- Each name must be between 2 and 32 characters in length.
- Monitoring server naming is case-sensitive on all platforms.

> **Note:** Use the following general steps to install the monitoring server. For detailed installation instructions, see *IBM Tivoli Monitoring Installation and Setup Guide*, GC32-9407, included in the product documentation and available at:
>
> http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/index.jsp?toc=/com.ibm.itm.doc/toc.xml

We describe the following installation steps in this section:

- Installing DB2 Express
- Installing Tivoli Monitoring Express V6.1 components

2.4.1 Installing DB2 Express

Perform the following tasks to install DB2 Express:

1. Log in to the system with the administrator account.
2. To start the installation, go to the installation image location from the IBM Tivoli Monitoring Express V6.1 CD 1. In our case, this is C:\itmexpress\disk1\.
3. Click **launchpad.exe**.

4. Click install **DB2 Express** from the Launchpad menu, as shown in Figure 2-4.

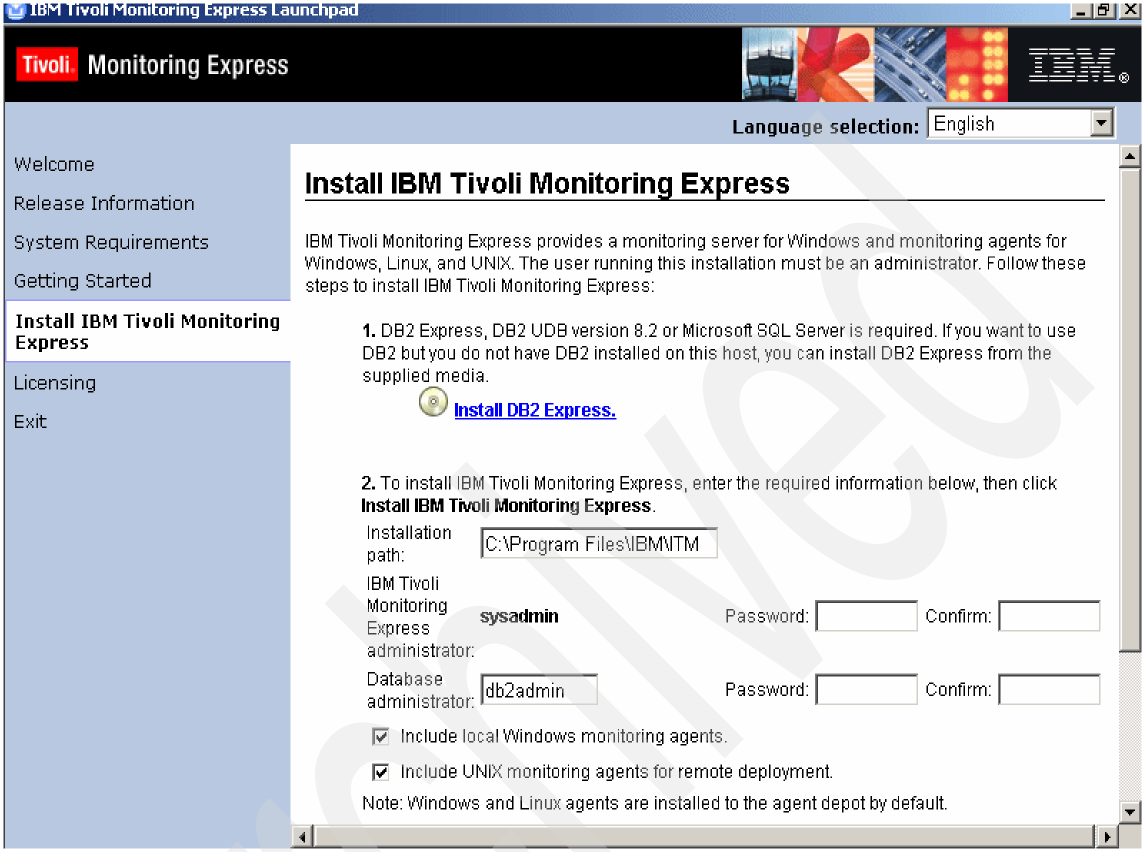


Figure 2-4 DB2 Express installation

5. Click **Next** in DB2 Express to start the DB2 Setup wizard.
6. Read the terms in the license agreement. Select **I accept the terms in the license agreement**, and click **Next**.

7. Select **Typical** and click **Next** under Select the installation type, as shown in Figure 2-5.

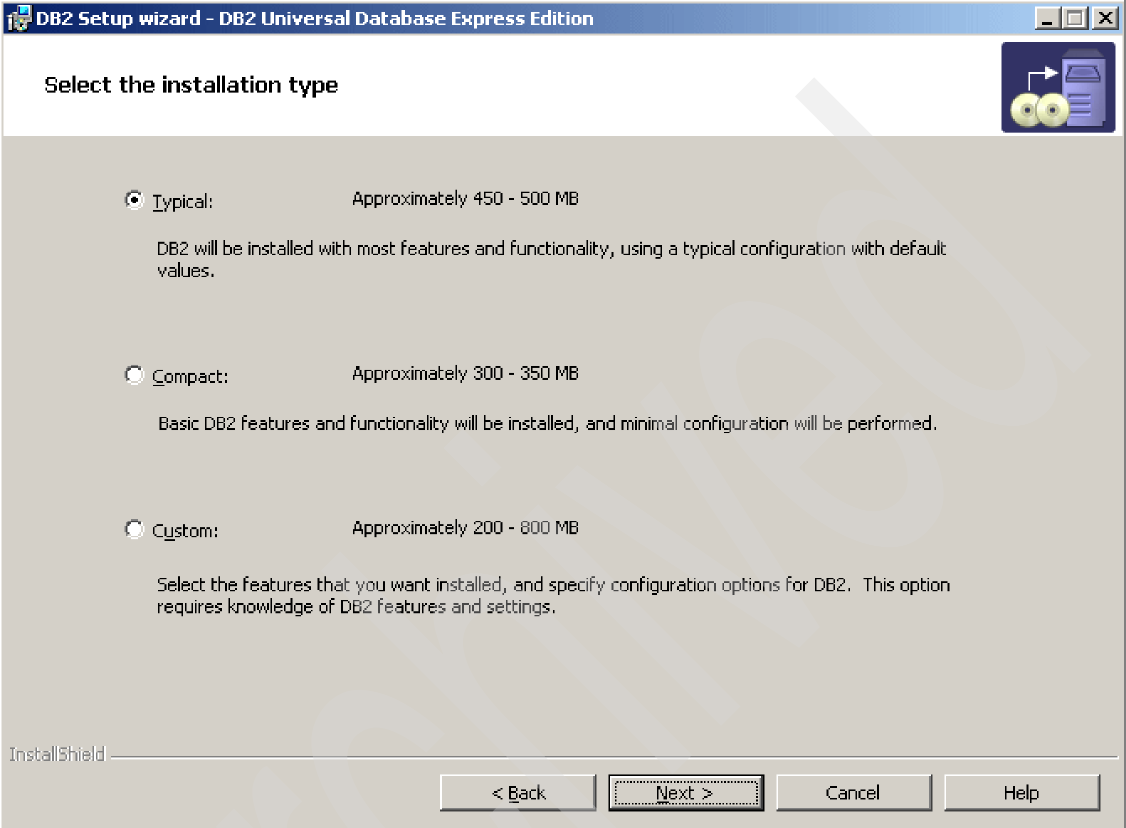


Figure 2-5 Selecting the installation type in the DB2 Setup wizard

8. Enter where the DB2 will be installed and click **Next**, as shown in Figure 2-6. In our case, we enter the `C:\IBM\SQLLIB` directory.

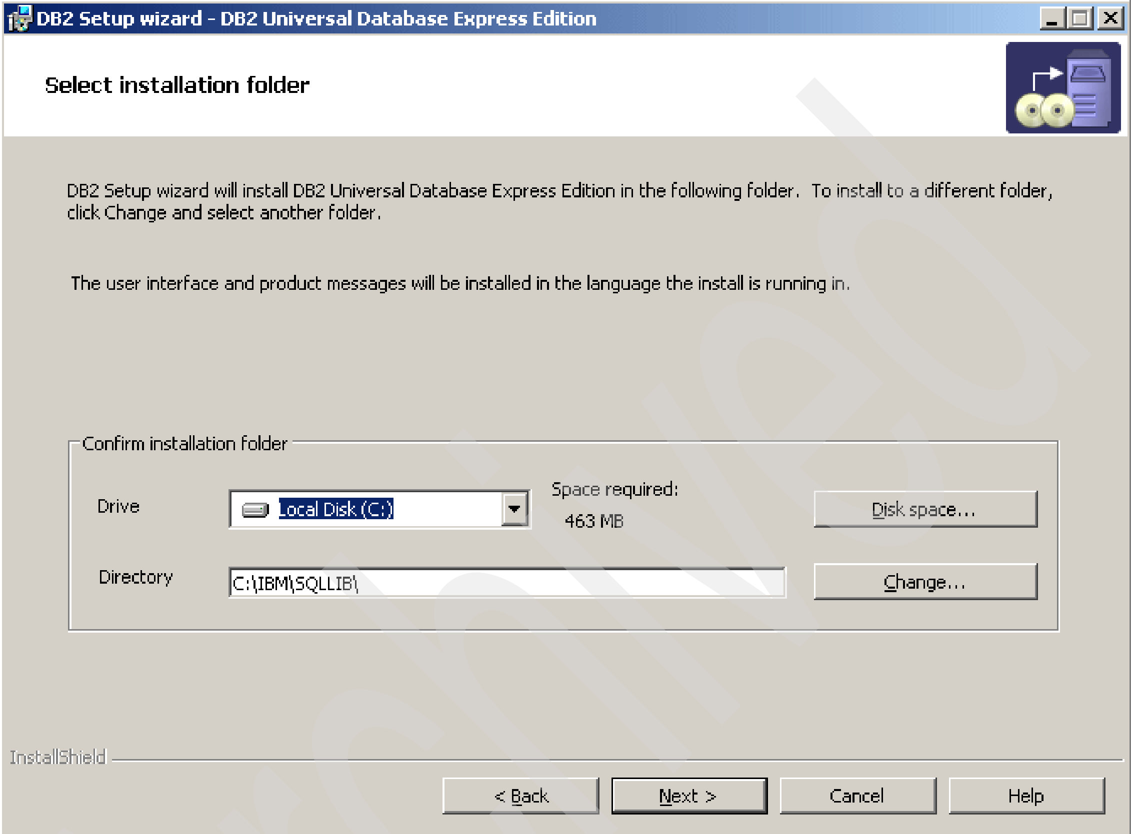


Figure 2-6 DB2 Express Setup wizard: Select installation folder

9. The DB2 Express Setup wizard creates a user for DB2 administration purposes. In the Set user information for the DB2 Administration Server window, displayed in Figure 2-7, select **Local user or Domain user account** and **Use the same user name and password for the remaining DB2 services**. In the User information area, enter the following details, and then click **Next**:
   - Domain: Leave this blank, unless you are using domain user.
   - User name: `db2admin`
   - Password: `db2admin`
   - Confirm password: `db2admin`

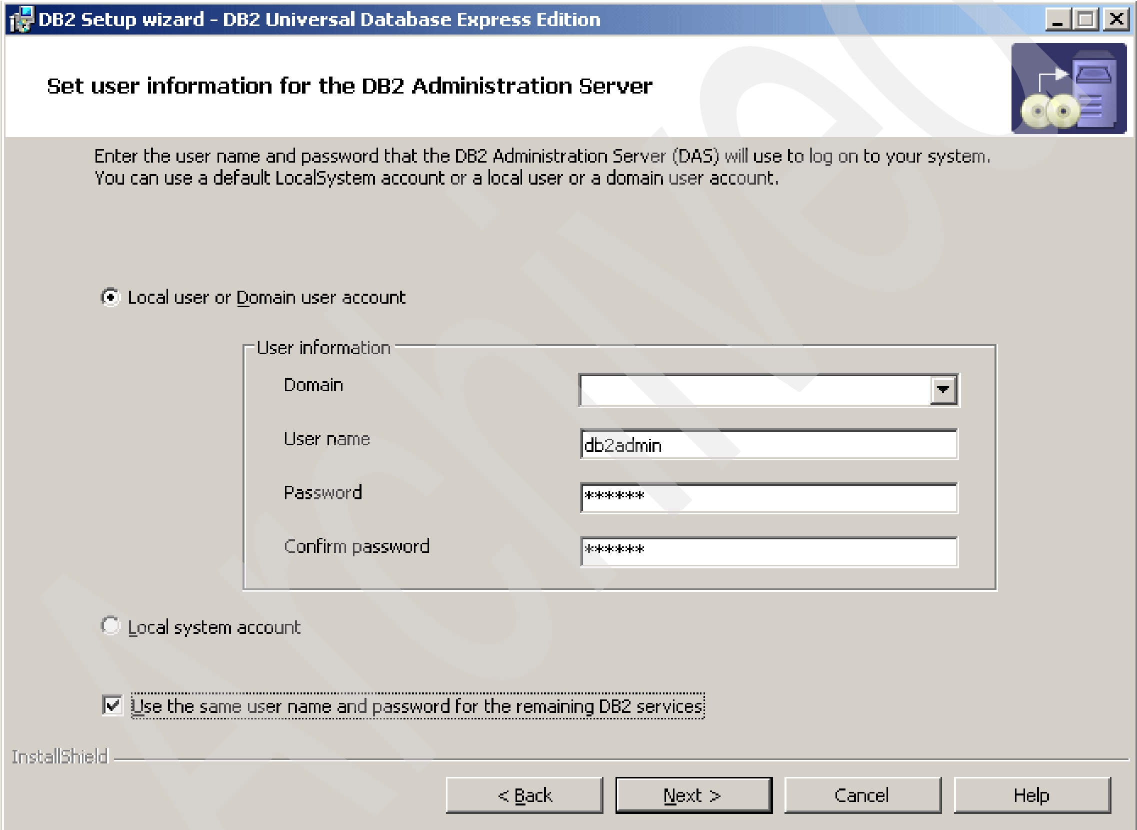


Figure 2-7 Setting user information for the DB2 Administration Server

10. Click **Next** in the Set up the administration contact list window. In our case, we did not configure it for this installation.

11. Click **Next** in the Configure DB2 instances window.

12.The last window shows the current settings. Click **Install** to start copying the files, as shown in Figure 2-8.

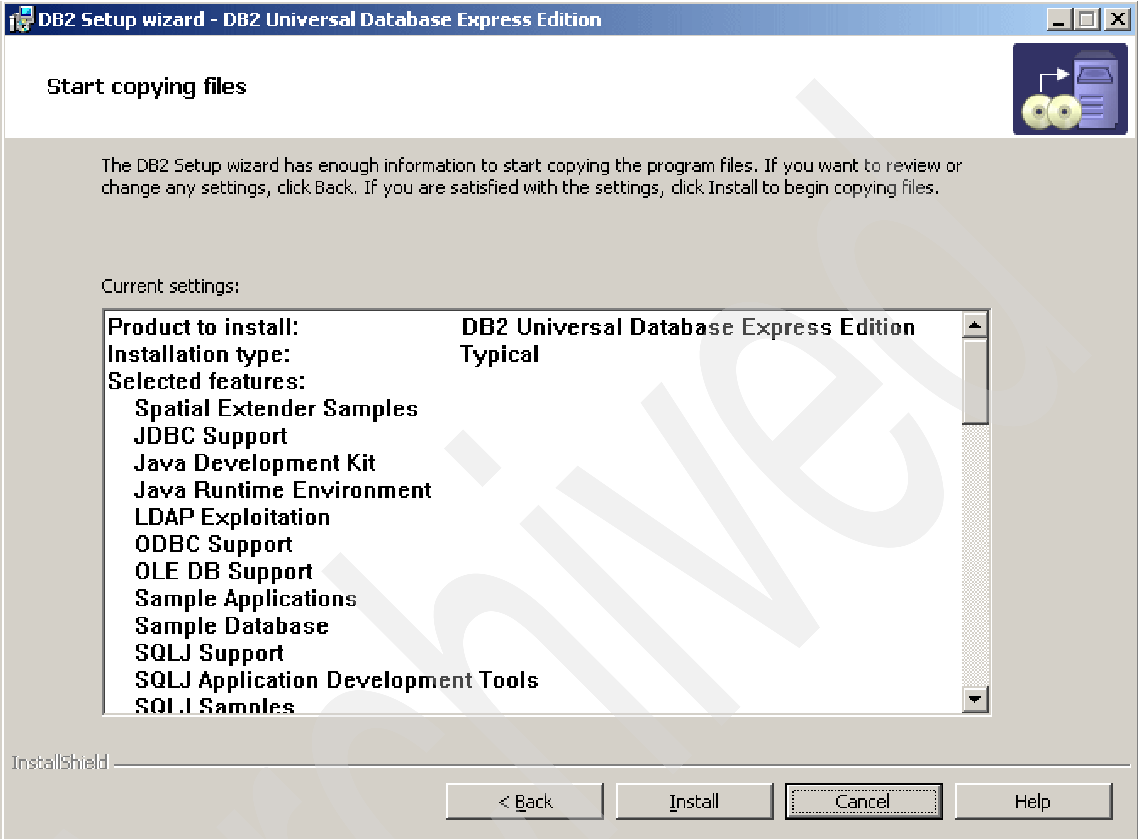


Figure 2-8 Copying files in DB2 Setup wizard

13.Click **Finish** to complete the DB2 Express installation.

> **Note:** After finishing the DB2 Express installation, the DB2 setup starts the IBM DB2 First Steps Launchpad and checks for DB2 updates. You can defer this task by clicking **No** and **Exit First Steps**.

After the installation, verify whether the database server is running by checking the DB2 services status by selecting **Start** → **Setting** → **Control Panel** → **Administrative Tools** → **Services**.

These services, started with DB2, should be running. Do not worry if the DB2 Governor service is turned off by default, as shown in Figure 2-9.

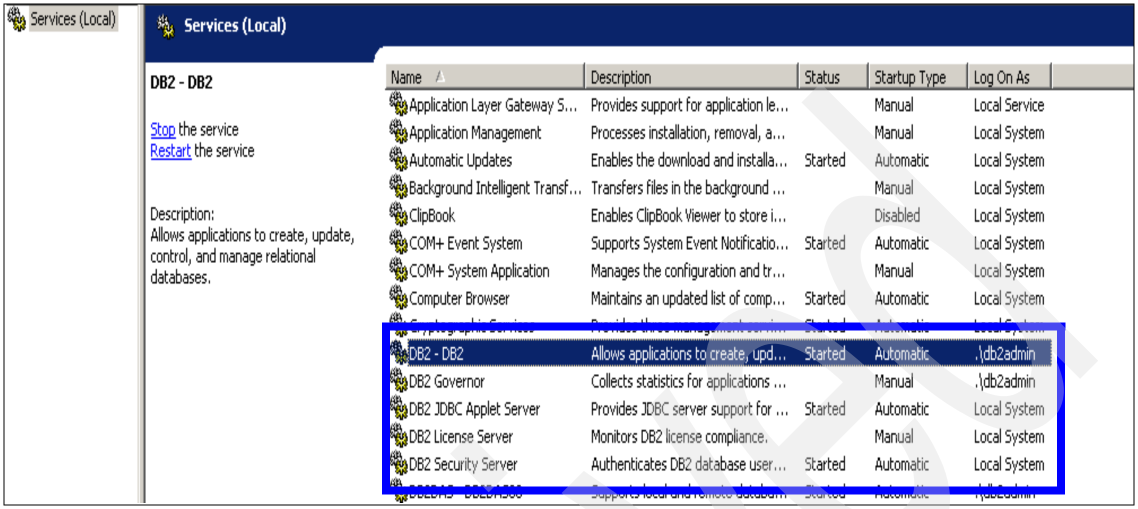


Figure 2-9 Services management console

2.4.2 Installing Tivoli Monitoring Express V6.1 components

When you install the IBM Tivoli Monitoring Express V6.1 server components using the IBM Tivoli Monitoring Express Launchpad, you have the option of installing some agents locally on the server.

The local agents monitor the server. In addition, the Launchpad creates an agent depot on the server and populates the depot with agent bundles or agent installation images that can be deployed to the remote computers from where you want to collect monitoring data.

Figure 2-10 on page 35 shows the Install IBM Tivoli Monitoring Express window of the Launchpad.

The Install IBM Tivoli Monitoring Express window of the Launchpad presents the following options for agents:

- Include local Windows monitoring agents.
- Include UNIX monitoring agents for remote deployment.

If you select the Include local Windows monitoring agents check box, the default agents, that is, the Windows OS agent, the Active Directory agent, the Universal Agent, the Warehouse Proxy agent, and the Warehouse Summarization and Pruning agent, will be installed on the server, that is, the Windows computer where you are running the Launchpad. These agents interact with the Windows OS, the Tivoli Data Warehouse component, and the Active Directory option. If you clear this check box, no agents are installed on the computer.

Regardless of which combination of check boxes you select or do not select, for example, if you do not select any check boxes, the Launchpad creates an agent depot on the server and populates it with the Windows and Linux agents from CD2. These agents are the Windows OS agent, the Universal Agent (Windows and Linux), the Active Directory agent, the Linux OS agent, and the UNIX logs.

If you select the Include UNIX monitoring agents for remote deployment check box, the Launchpad adds UNIX agents to the agent depot from CD3. These agents are the UNIX OS agent, the Universal Agent (UNIX), and the UNIX logs agent.

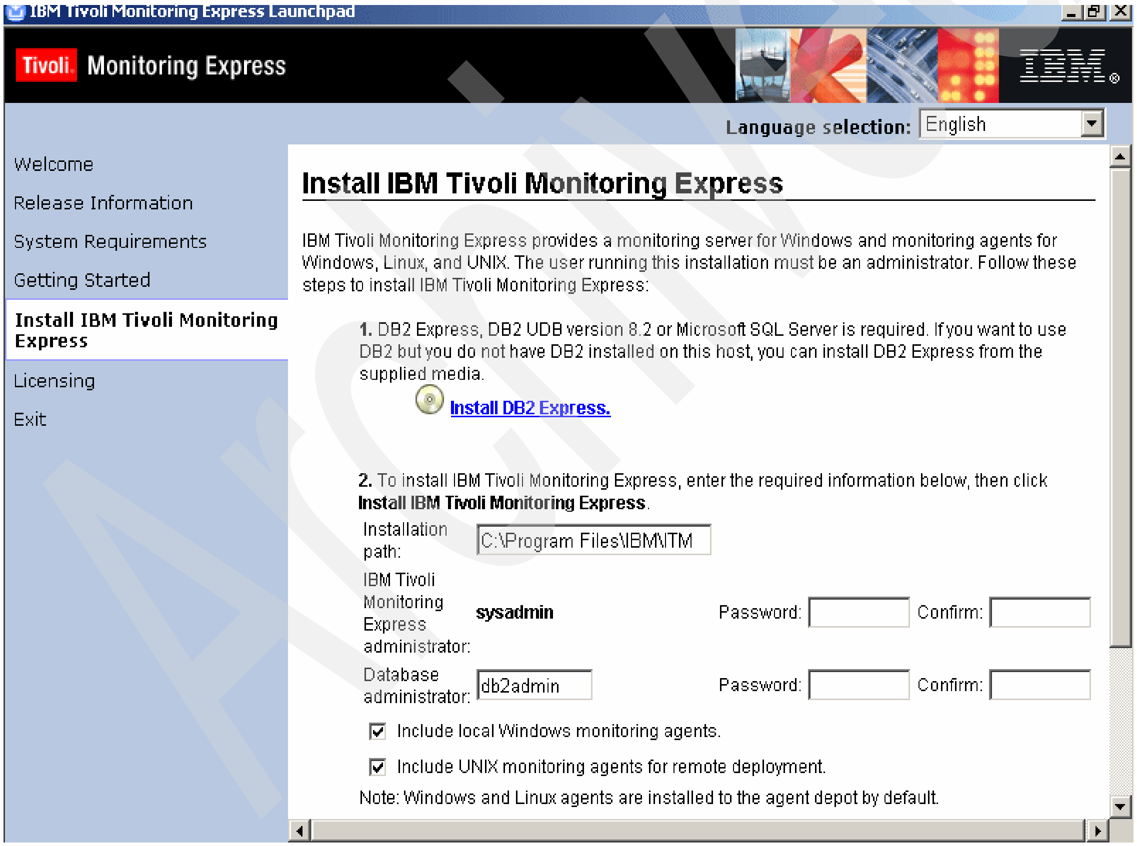


Figure 2-10 IBM Tivoli Monitoring Express installation

To install IBM Tivoli Monitoring Express V6.1, perform the following tasks in the window displayed in Figure 2-11:

1. Type the path where you want to install IBM Tivoli Monitoring Express V6.1.
2. The IBM Tivoli Monitoring Express user ID is sysadmin. Type the password of your choice.
3. The DB2 Express user ID *and* the password is db2admin. Enter these values.

> **Note:** All the passwords should be compliant with your local Windows policy and Active Directory policy.

4. By default, both the check boxes on the Install IBM Tivoli Monitoring Express window are selected so that the Launchpad installs agents locally on the server and populates the agent depot with all the agents available for remote deployment.
5. Select the license agreement check box.
6. Click **Install IBM Tivoli Monitoring Express.** When prompted, insert the IBM Tivoli Monitoring Express V6.1 CD2 and CD3 to install the product.

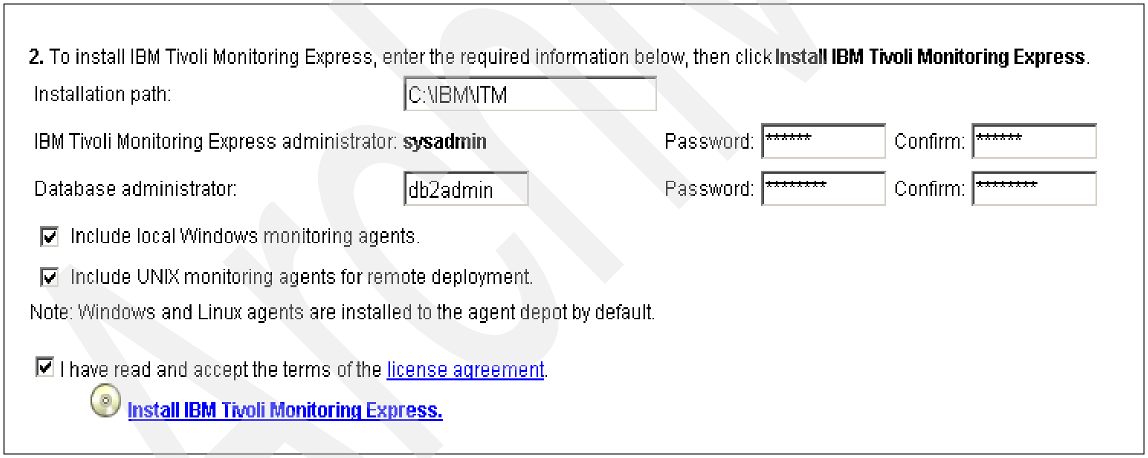


Figure 2-11 IBM Tivoli Monitoring Express options

7. Click **OK** after the installation finishes.

To verify if the installation is running well, click **Launch Tivoli Enterprise Portal** from the icon on the desktop. Use the user ID and password referenced in step 2 to log in to Tivoli Enterprise Portal, as shown in Figure 2-12 on page 37.

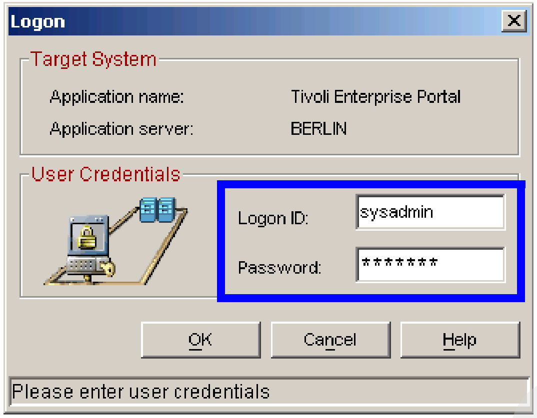


Figure 2-12 Logon window

Launching Tivoli Enterprise Portal

You can access Tivoli Enterprise Portal in two ways, using a *browser* or a *desktop client*.

From a functional point of view, there are no differences between the two. However, the browser-based client has two advantages:

- You do not have to install an updated client if a new version is available. The browser client will always be at the latest level available from the server.
- 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 that you lose desktop space due to the headers of the Web browser.

Launching Tivoli Enterprise Portal from a desktop client

To launch the Desktop client (Figure 2-13), select **Start** → **Programs** → **IBM Tivoli Monitoring** → **Tivoli Enterprise Portal**.

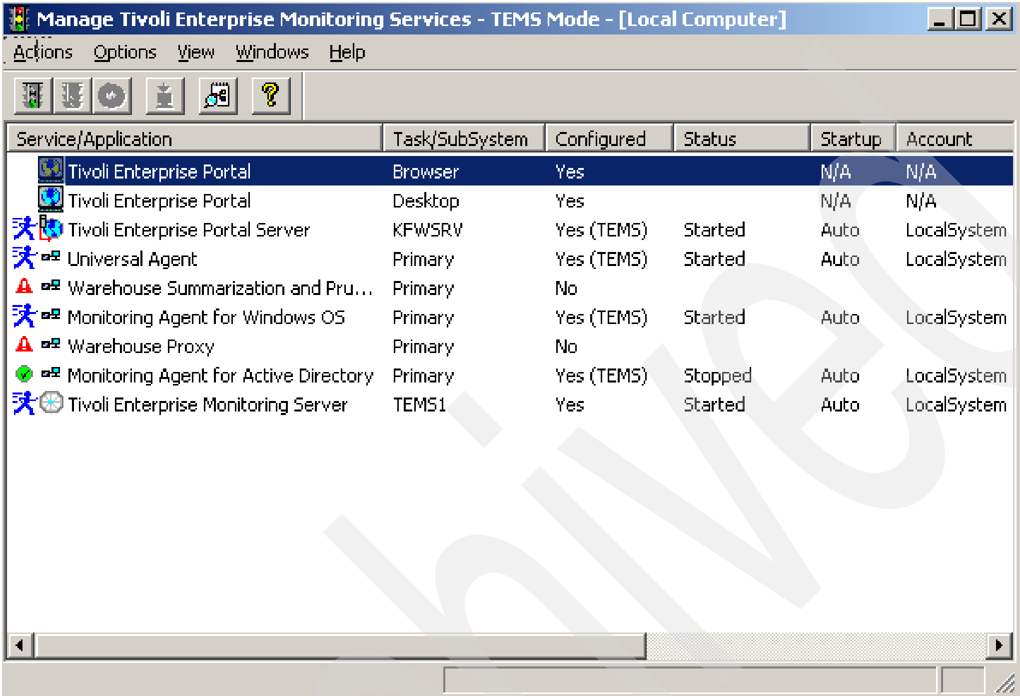


Figure 2-13 Tivoli Enterprise Portal client desktop

Launching Tivoli Enterprise Portal from a browser

To launch Tivoli Enterprise Portal using a browser, perform the following tasks:

1. Select **Start** → **Programs** → **Internet Explorer**.
2. Type the following URL, where *hostname* is the host name of the Tivoli Enterprise Portal Server:

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

3. Click **Yes** to accept the Warning - Security message shown in Figure 2-14.

Figure 2-14 The security certificate message

4. In the Logon window (Figure 2-15), enter `sysadmin` in the User Credentials area for Logon ID and your password for the Password field. Click **OK**.

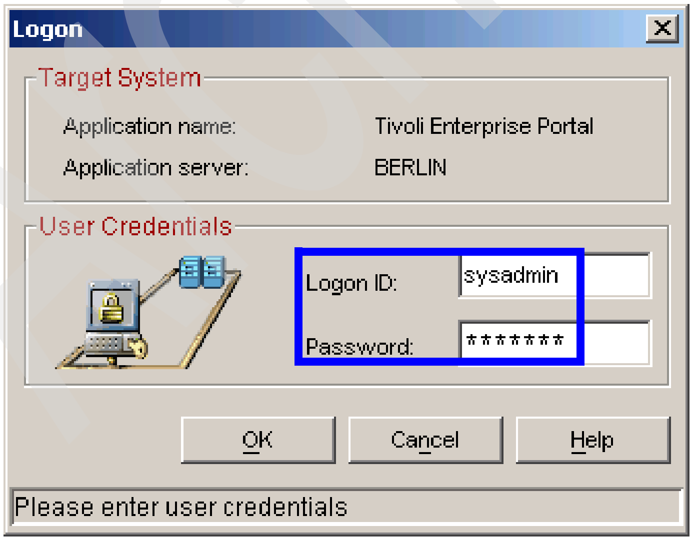


Figure 2-15 Logon window

5. In the Security Alert window (Figure 2-16), click **Always Accept** to accept the certificate.

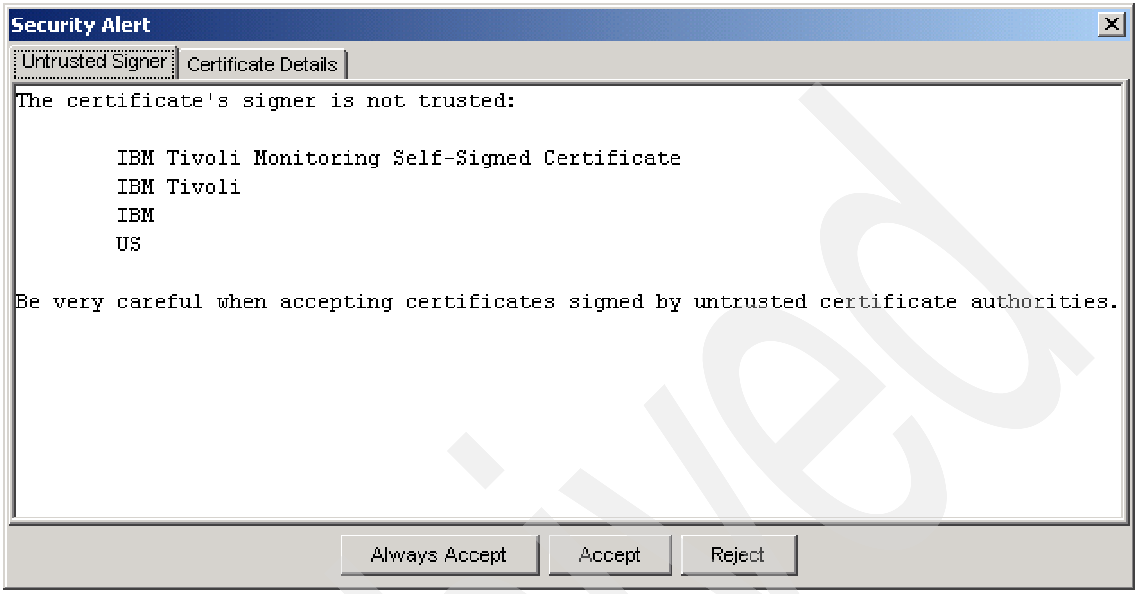


Figure 2-16 Security Alert

There are two Internet Explorer windows: Welcome to IBM Tivoli Monitoring and Tivoli Enterprise Portal. Because you have not configured and installed any agents, you can only see the Enterprise Navigator, with no agent running.

6. Click **Exit** to close the Welcome to IBM Tivoli Monitoring window, and select **File** → **Exit** → **Yes** to close Tivoli Enterprise Portal. You can close Internet Explorer, too.

7. After you start the browser client, change the memory settings for the Java plug-in used by Tivoli Enterprise Portal:

   a. Open the Windows Control Panel and double-click the Java plug-in used by Tivoli Enterprise Portal.

b. In the Advanced page, enter `-Xms64m -Xmx256m` in the Java Runtime Parameters field and click **Apply**, as shown in Figure 2-17.

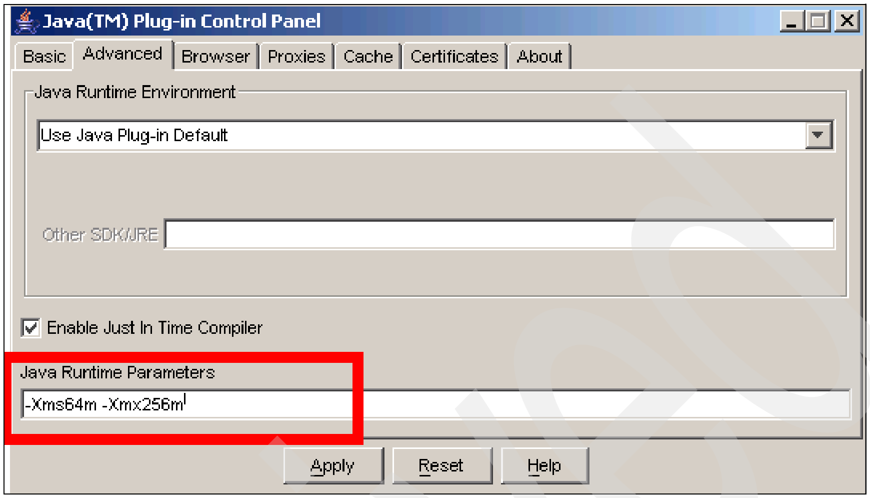
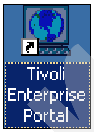


Figure 2-17 Java Runtime Parameters

c. Log off the portal and then log in again.

Launching Tivoli Enterprise Portal client desktop application

You can launch the desktop client using **Start** → **Programs** → **IBM Tivoli Monitoring** → **Tivoli Enterprise Portal** or by launching its icon on the desktop.



2.5 Configuring IBM Tivoli Monitoring Express V6.1 components

This section describes the processes involved in configuring the following IBM Tivoli Monitoring Express V6.1 components:

- Configuring Tivoli Enterprise Monitoring Server
- Configuring Tivoli Enterprise Portal Server
- Configuring the Warehouse Proxy agent
- Configuring the Summarization and Pruning agent
- Configuring hot standby Tivoli Enterprise Monitoring Server

By default, the Tivoli Enterprise Portal Server component is offline.

2.5.1 Stopping and starting Tivoli Monitoring Express V6.1 components

IBM Tivoli Monitoring Express V6.1 provides an application called Manage Tivoli Enterprise Monitoring Services that you can use to stop and start components.

Manage Tivoli Enterprise Monitoring Services is on any machine that has IBM Tivoli Monitoring Express V6.1 installed on it, including agents. Manage Tivoli Enterprise Monitoring Services includes the following functions:

- Configure Tivoli Enterprise Monitoring Server.
- Configure OS agents, Universal Agents, and application agents.
- Configure the Warehouse Summarization and Pruning agent.
- Manage log files.
- Stop and start the agents, the Tivoli Enterprise Monitoring Server, and the Tivoli Enterprise Portal Server.

From the Manage Tivoli Enterprise Monitoring Services console, you can:

- See the services status.
- Start, stop, and recycle the services.
- Change the startup.
- Configure and reconfigure the services.
- Launch Tivoli Enterprise Portal.
- Edit the ENV file.
- View the trace log.
- Edit the trace parameters.

The following exercise shows you how to use this application to manually stop and start servers and agents:

1. On the Windows system where you installed the product, select **Start** → **Programs** → **IBM Tivoli Monitoring** → **Manage Tivoli Monitoring Services**. The Manage Tivoli Enterprise Monitoring Services window opens, as shown in Figure 2-18 on page 43. The symbol next to a service indicates its current state:
   - A blue running figure indicates that the service is started.
   - A green check mark indicates that the service is configured and can be started.
   - A red exclamation mark indicates that the service needs to be configured before it can be started.

   The effect of double-clicking a service depends on its current state:

    For a service that is not yet configured, double-clicking launches the configuration menu.

For a configured service that is not yet started, double-clicking starts the service.

For a running service, double-clicking stops the service.

2. Highlight all the servers and agents and click the **Stop** button (red traffic light) on the toolbar (Figure 2-18).

3. Verify that all components are configured. A green check mark is displayed next to all configured components (Figure 2-18).

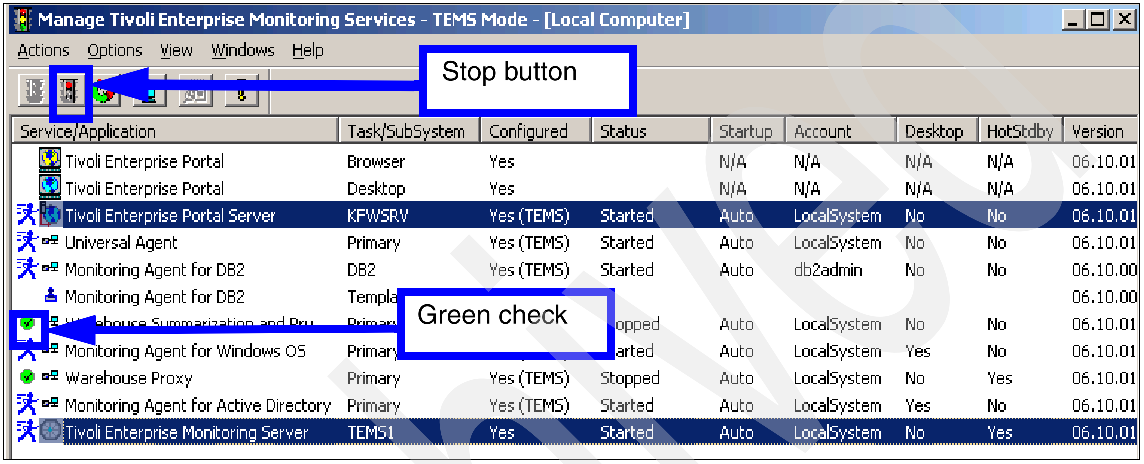


Figure 2-18 Manage Tivoli Enterprise Monitoring Services

4. Start all the components. Use one of the following different methods available to start a configured component:

   – Double-click **Tivoli Enterprise Monitoring Server** to start the monitoring server.

– Right-click **Tivoli Enterprise Portal Server** and click **Start** to start the portal server, as shown in Figure 2-19.

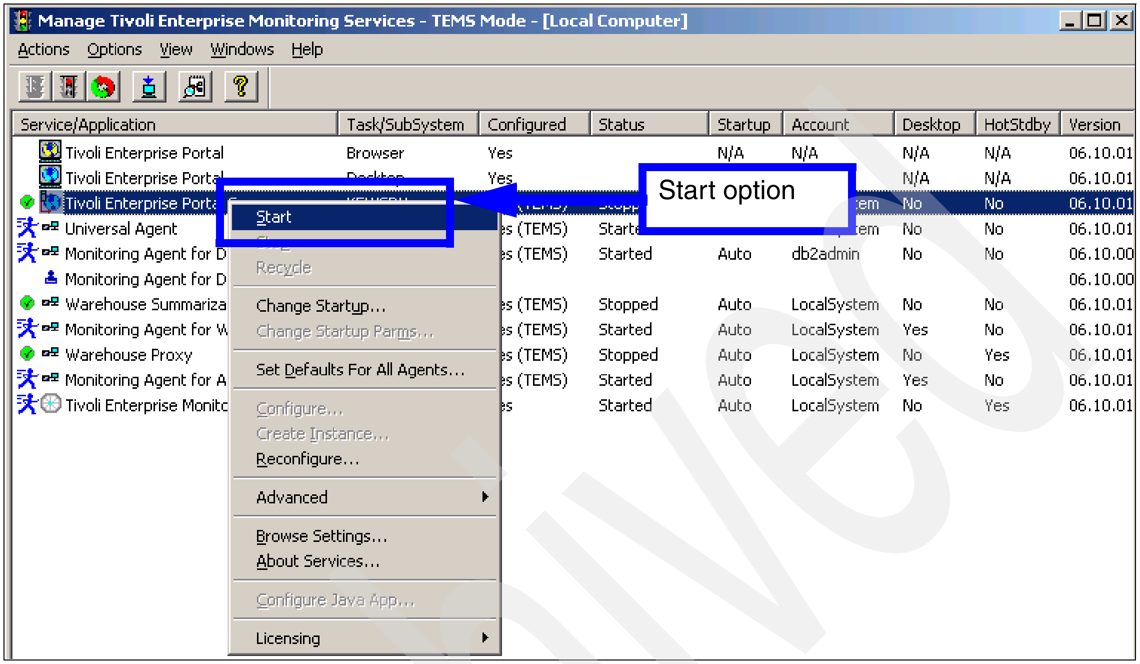


Figure 2-19 Start Tivoli Enterprise Portal Server

– Select all configured agents and click the **Start** button (green traffic light) on the toolbar, as shown in Figure 2-20.

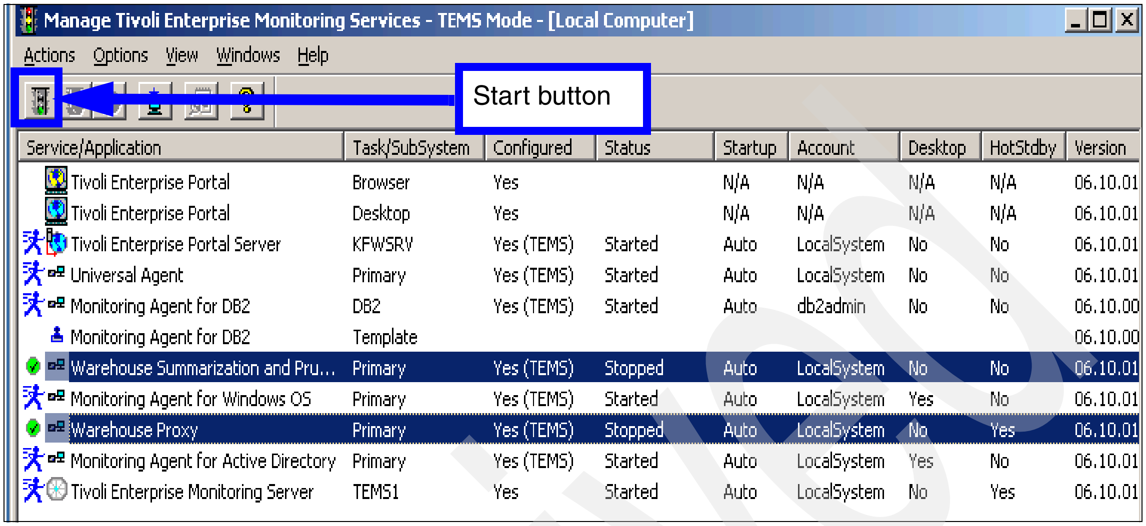


Figure 2-20 Start Warehouse Proxy agent and Summarization and Pruning agent

5. Use any of the methods described to start the Tivoli Enterprise Portal desktop client.

You can manually start or stop each service that is installed on a Windows system using the application. You can also specify whether or not the service should start automatically when the system boots, as shown in Figure 2-21 on page 46.

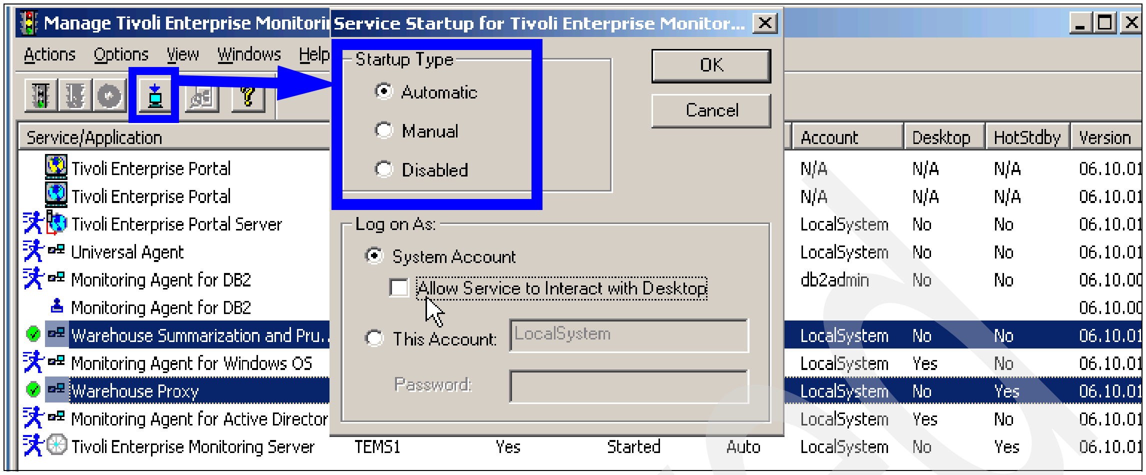


Figure 2-21 Startup options

After an agent successfully connects to the hub Tivoli Enterprise Monitor Server, it registers with it and is visible in Tivoli Enterprise Portal as one or multiple managed systems.

2.5.2 Configuring Tivoli Enterprise Monitoring Server

To configure the Tivoli Enterprise Monitoring Server, perform the following tasks:

1. Launch the Tivoli Enterprise Monitoring Services window displayed in Figure 2-22 by selecting **Start** → **Programs** → **IBM Tivoli Monitoring** → **Manage Tivoli Monitoring Services**.

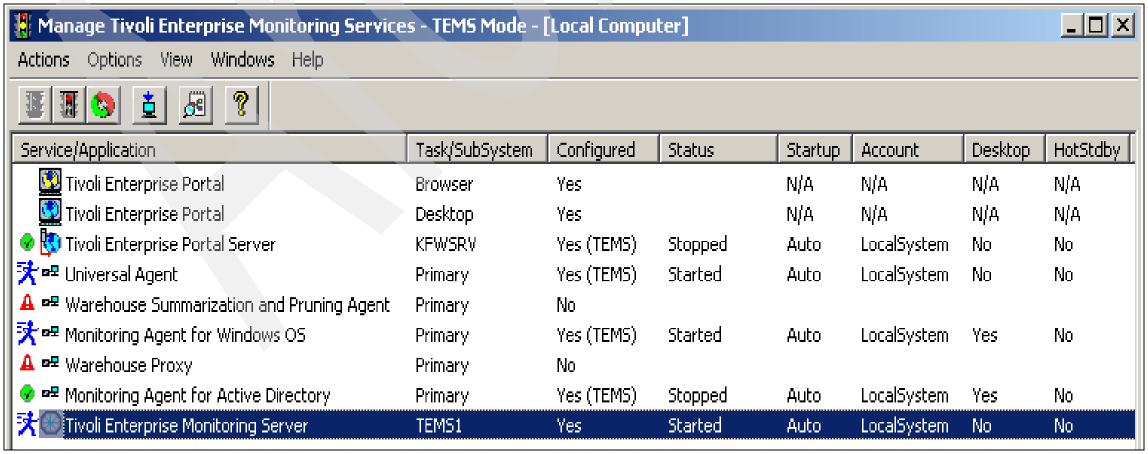


Figure 2-22 Tivoli Enterprise Monitoring Services

2. Right-click the **Tivoli Enterprise Monitoring Server** service and select **Reconfigure**.
3. Clear the **Security: Validate User** option. Then log in with a sysadmin account without any password. Click **OK**.

> **Note:** By default, **IP.PIPE** is specified for Protocol 1, which uses unsecured TCP communications, as shown in Figure 2-23.

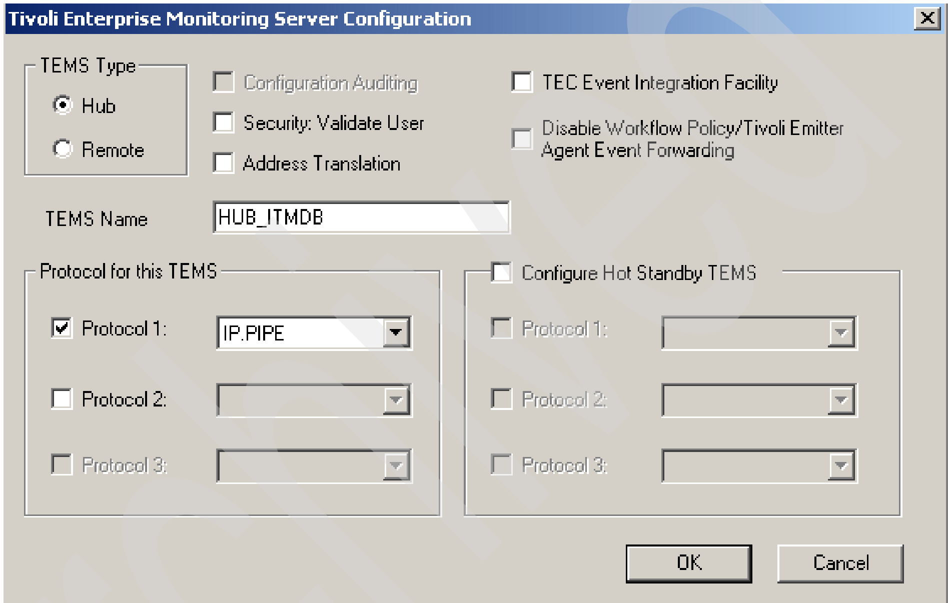


Figure 2-23 Tivoli Enterprise Monitoring Server Configuration

4. Click **OK** to accept the default settings of Tivoli Enterprise Monitoring Server, as shown in Figure 2-24.

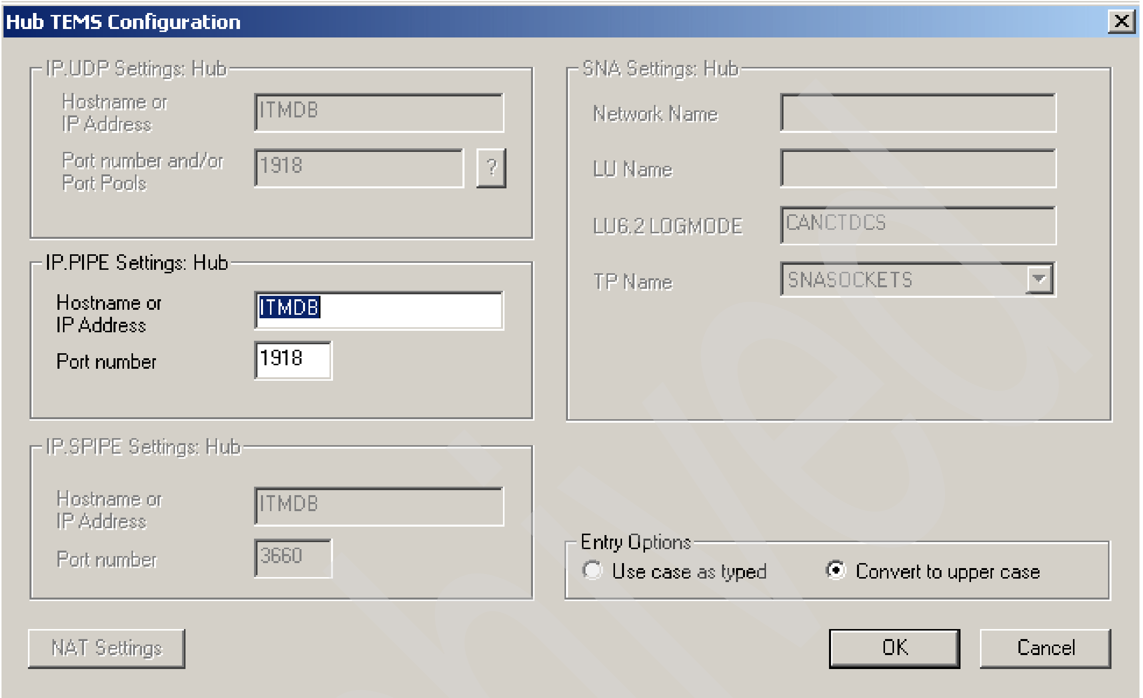


Figure 2-24 Tivoli Enterprise Monitoring Server Configuration

The Tivoli Enterprise Monitoring Server component is now configured.

2.5.3 Configuring Tivoli Enterprise Portal Server

To configure the Tivoli Enterprise Portal Server, perform the following tasks:

1. Go to the Tivoli Enterprise Monitoring Services window displayed in Figure 2-22 on page 46.
2. Right-click the **Tivoli Enterprise Portal Server** service and select **Reconfigure**.

3. Leave the defaults parameters. Clear the **Connection must pass through firewall** option. Protocol1 should be **IP.PIPE** (Figure 2-25).

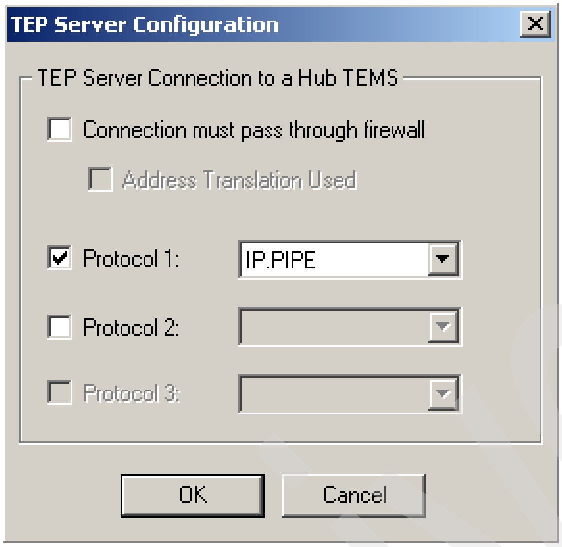


Figure 2-25 Tivoli Enterprise Portal Server Configuration

> **Note:** You can specify the following communications parameters:
>
> - **IP.PIPE**: This uses unsecured TCP communications.
> - **IP.SPIPE**: This uses SSL secure TCP communications.
> - **SNA**: This uses SNA for mainframe environments.
> - **IP.UDP**: This uses unsecured UDP communications.

4. Click **OK** to accept the default settings of Tivoli Enterprise Portal Server (Figure 2-26).

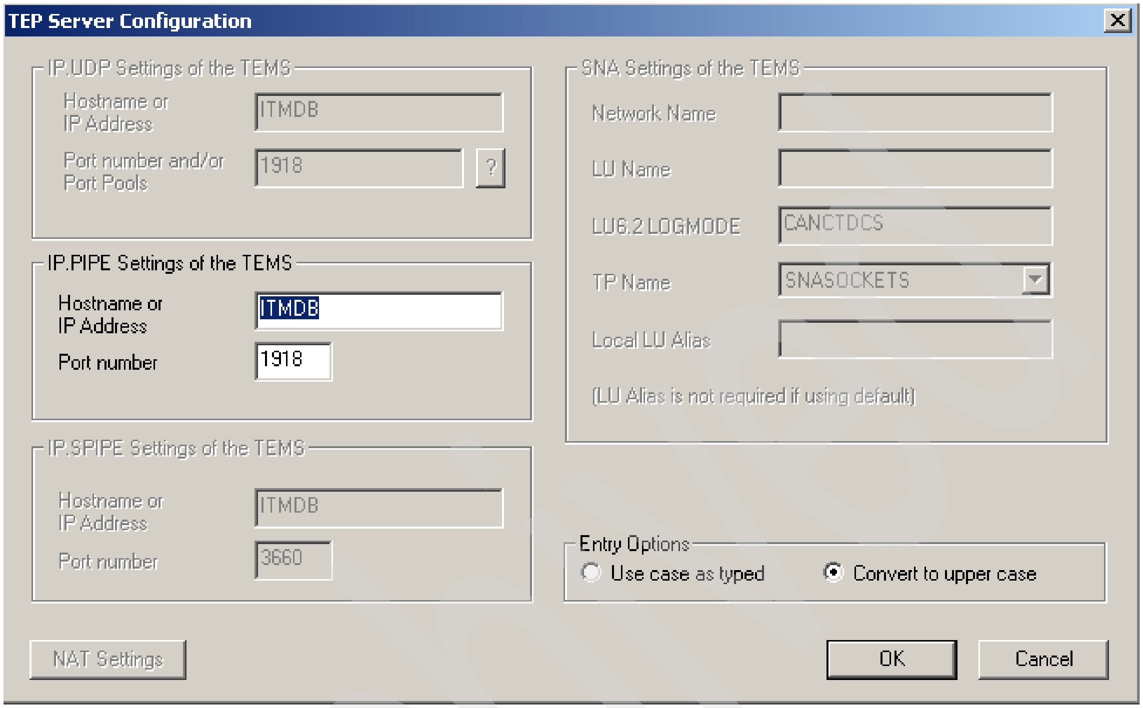


Figure 2-26 Tivoli Enterprise Portal Server Configuration

5. Click **Yes** when you are asked whether you want to reconfigure the warehouse connection information for the Tivoli Enterprise Portal Server (Figure 2-27).

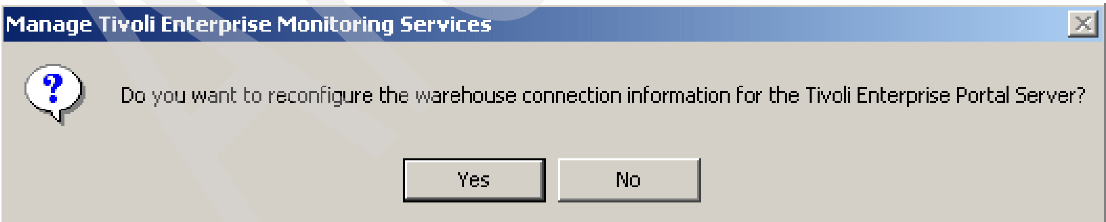


Figure 2-27 Reconfiguring warehouse connection information

6. Select **DB2** for the Warehouse Proxy Database Selection (Figure 2-28).

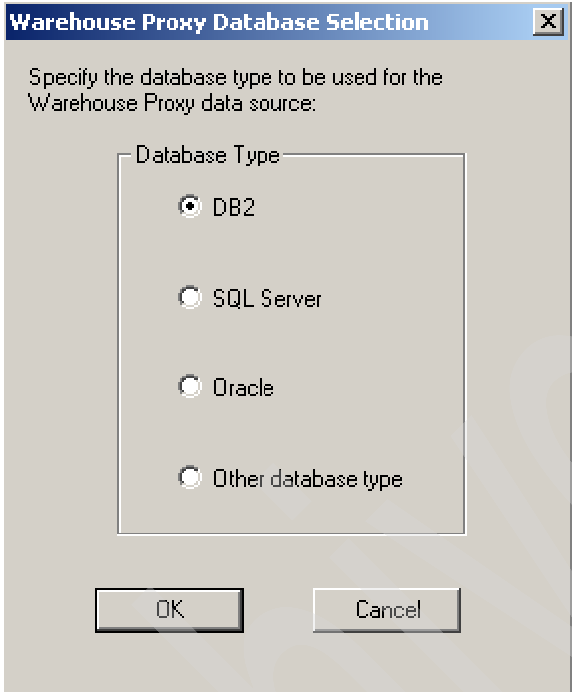


Figure 2-28 Selecting the Warehouse Proxy Database

7. Enter the following information in the Configure DB2 Data Source for Warehouse Proxy window displayed in Figure 2-29, and click **OK**:
   - Data Source Name: `TM Warehouse`
   - Database Name: `WAREHOUS`
   - Admin User ID: `db2admin`
   - Admin Password: `db2admin`
   - Database User ID: `ITMUser`
   - Database Password: `itm61expr`
   - Reenter Password: `itm61expr`

   > **Note:** By default, ITMUser is created during the installation of IBM Tivoli Monitoring Express V6.1.

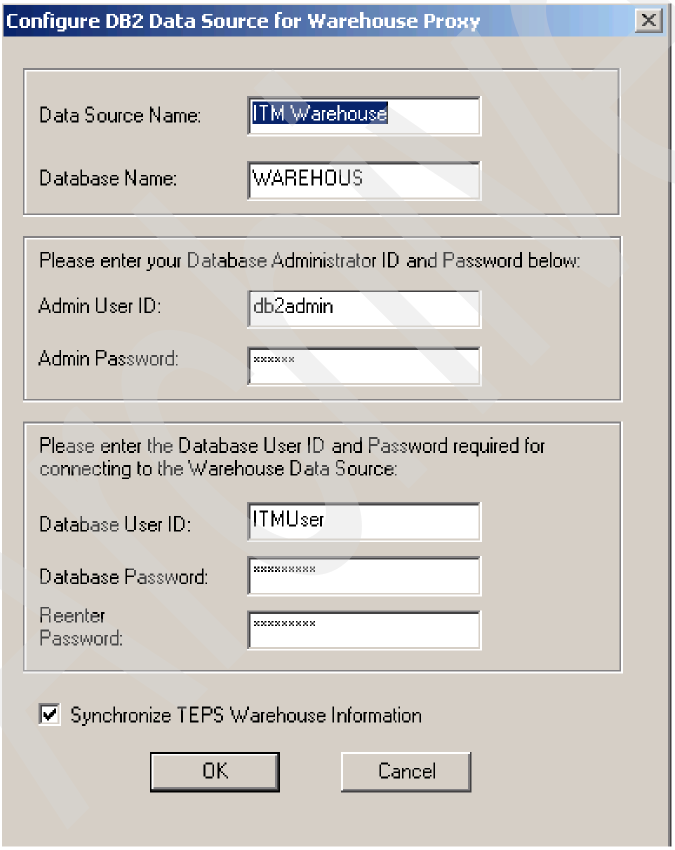


Figure 2-29 Configure DB2 Data Source for Warehouse Proxy

8. Click **OK** to finish the warehouse data source configuration.

   The Manage Tivoli Enterprise Monitoring Services - TEMS Mode - [Local Computer] window opens, as shown in Figure 2-30.

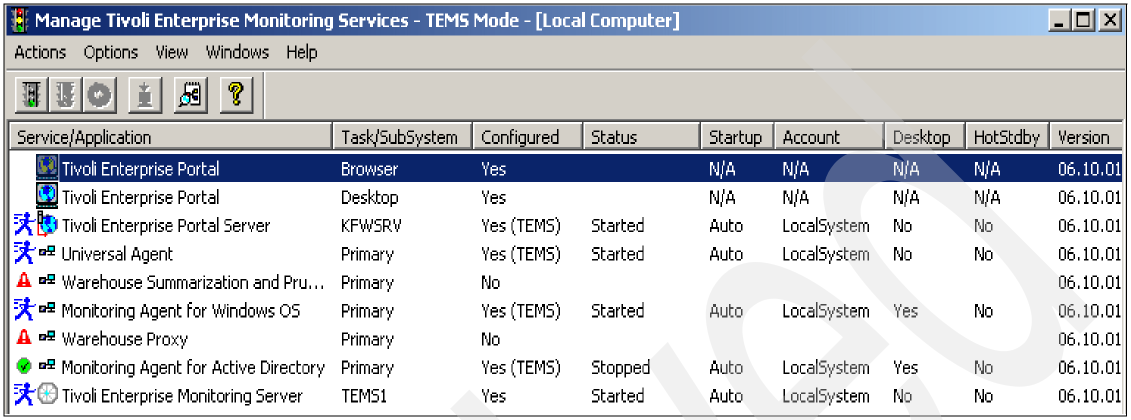


Figure 2-30 Manage Tivoli Enterprise Monitoring Services

9. After the configuration is complete, restart the Tivoli Enterprise Monitoring Server and the Tivoli Enterprise Portal Server services through the management console, as shown in Figure 2-31.

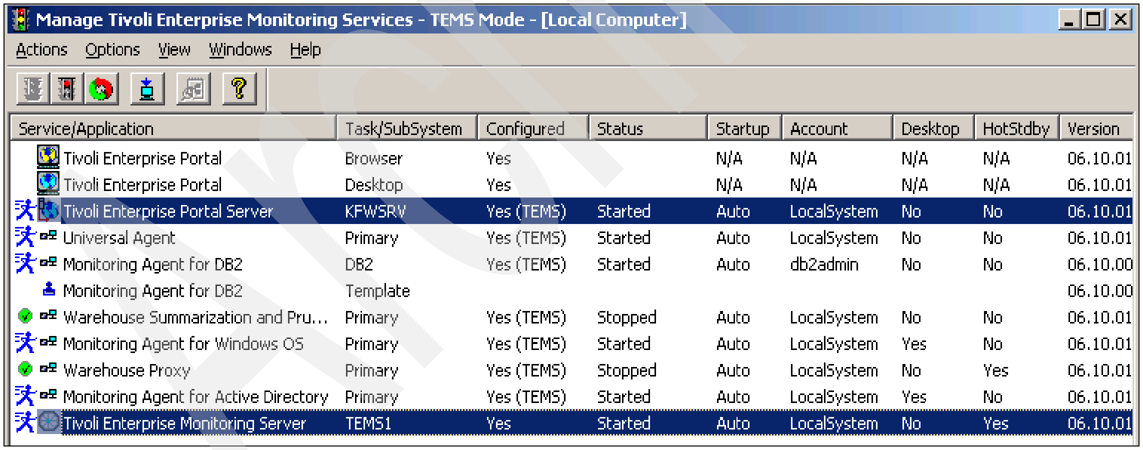


Figure 2-31 Manage Tivoli Enterprise Monitoring Services window

2.5.4 Configuring the Warehouse Proxy agent

The Warehouse Proxy agent transfers data from monitoring agents or the monitoring server to the Tivoli Data Warehouse database. The Tivoli Data

Warehouse is the component of IBM Tivoli Monitoring Express V6.1 that stores historical data. The Tivoli Data Warehouse database is an Open Database Connectivity (ODBC) data source that is created in the DB2 application when you install IBM Tivoli Monitoring Express V6.1.

The procedure described here shows you how to configure the Warehouse Proxy agent to connect to the Tivoli Data Warehouse database that is created during the installation. If you require a large database to maintain historical data, follow the database planning and configuration procedures described in the IBM Redbook *Getting Started with IBM Tivoli Monitoring 6.1 on Distributed Environments*, SG24-7143.

> **Note:** Although a green check mark is displayed next to the Warehouse Proxy agent after you install IBM Tivoli Monitoring Express V6.1, the agent is, in fact, not configured. Continue the procedure described here to configure the agent.

Before beginning the process, check whether the Warehouse Proxy has the correct connection parameters for the Warehouse:

1. In the Manage Tivoli Enterprise Monitoring Services window, right-click **Warehouse Proxy** and select **Reconfigure**. You will see a warning stating that the Warehouse Proxy should to a HUB TEMS. Acknowledge this by clicking **OK**.
2. Accept the defaults in the windows that follow until you are asked whether you want to configure the ODBC data source, as shown in Figure 2-32. When asked this question, click **Yes**. This takes you to the window shown in Figure 2-33 on page 55.

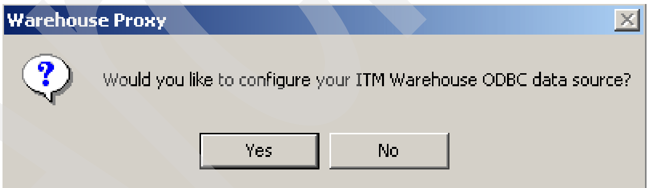


Figure 2-32 Warehouse Proxy window

3. Select **DB2** as the database, as shown in Figure 2-33, and click **OK**.

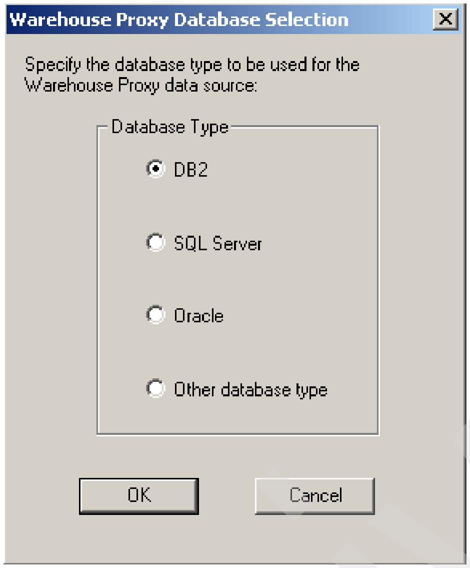


Figure 2-33 Selecting the Warehouse Proxy Database

4. In the ODBC configuration window, make sure that you have `WAREHOUS` as the Database Name, `db2admin` as the Admin User ID and Password, and `ITMUser` as the Database User ID and `itm61expr` for the Password, as shown in Figure 2-34.

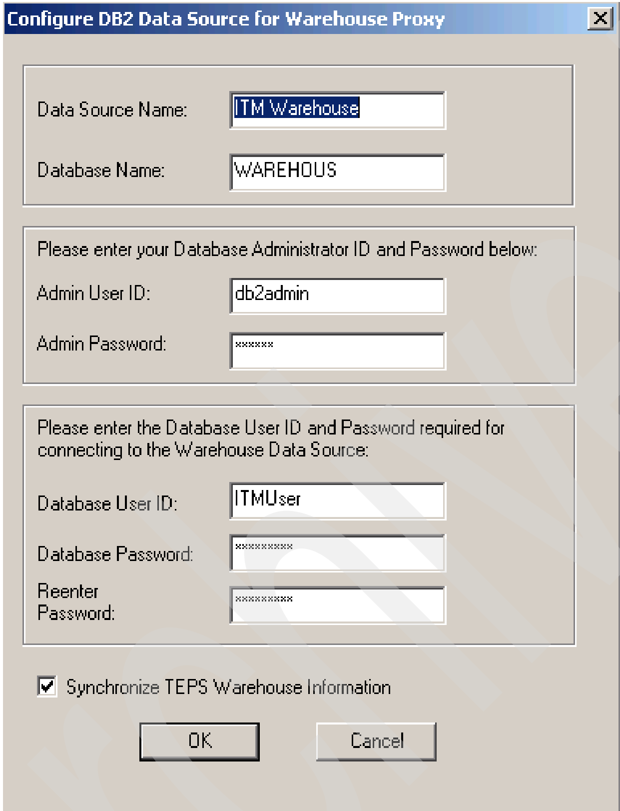


Figure 2-34 Configure DB2 Data Source for Warehouse Proxy

> **Note:** The Synchronize TEPS Warehouse Information check box *should be* selected (as shown in Figure 2-34). Otherwise, you have to configure this again separately for Tivoli Enterprise Portal Server. After completing this step, the database and the associated tables are created on your database.

5. Click **OK** when done.
6. This stops the Warehouse Proxy service. Double-click it to start again.

We are now ready to start configuring data collection and warehousing.

Important: You can change the default ODBC data source name using the following procedure:

1. Edit the Windows registry with the following command:

```
regedit
```

2. Find the following 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.

2.5.5 Configuring the Summarization and Pruning agent

The Summarization and Pruning agent is responsible for aggregating historical data and pruning the size of the database according to the desired guidelines.

Perform the following tasks to configure the agent:

1. In the Manage Tivoli Enterprise Monitoring Services window (Figure 2-31 on page 53), right-click **Warehouse Summarization and Pruning Agent** and select **Reconfigure**.
2. By default, the Connection must pass through firewall option is selected. Clear this option and click **OK**, as shown in Figure 2-35.

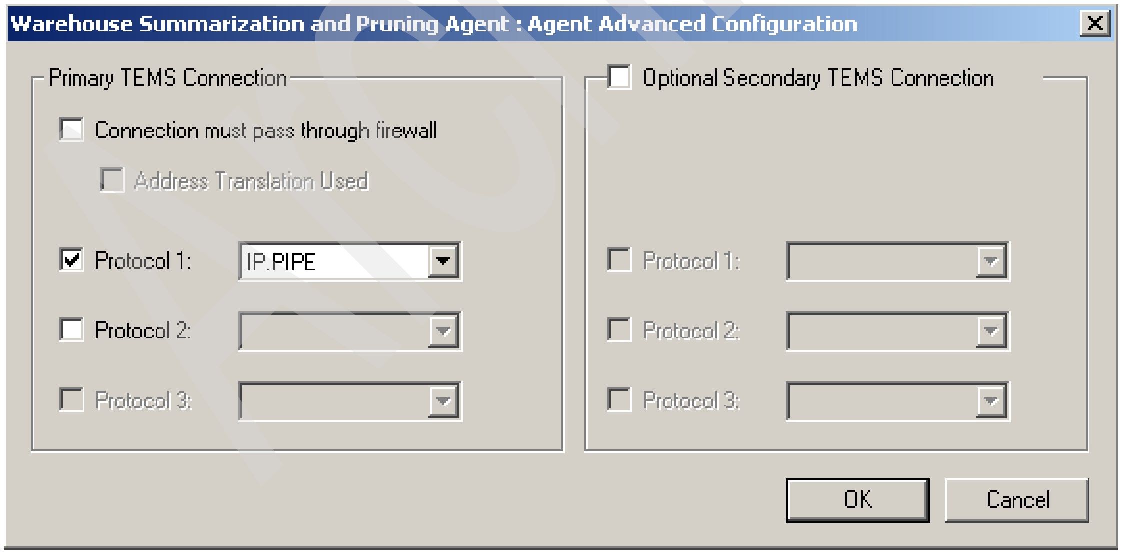


Figure 2-35 Warehouse Summarization and Pruning Agent: Agent Advanced Configuration

3. Accept the next configuration window with **OK**.
4. You will be asked whether you want to configure the Summarization and Pruning agent. Click **Yes**, as shown in Figure 2-36.

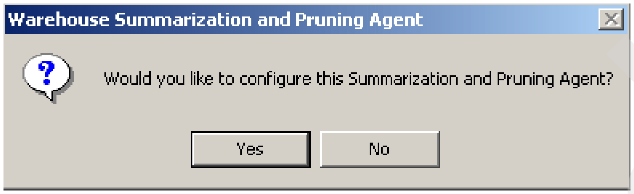


Figure 2-36 Warehouse Summarization and Pruning Agent configuration

5. The configuration window displayed in Figure 2-37 on page 59 opens after a few seconds. This shows the configuration settings to get to the Data Warehouse. Leave these settings as they are. On the Sources tab, enter the Tivoli Data Warehouse database and Tivoli Enterprise Portal Server information. Before performing any updates, confirm that the default configuration is accurate.

Note: If your Tivoli Data Warehouse database is on Microsoft SQL Server, install the JDBC™ drivers from the Microsoft SQL Server Web site. You need the following three files:

- 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.

From the pull-down list, select the database type for your Tivoli Data Warehouse.

While configuring the Warehouse Proxy, a database user called ITMUser is created by default. The user ID that you enter here must match that database user.

6. Click **Test database connection** to ensure that you can communicate with your Tivoli Data Warehouse database (Figure 2-37).

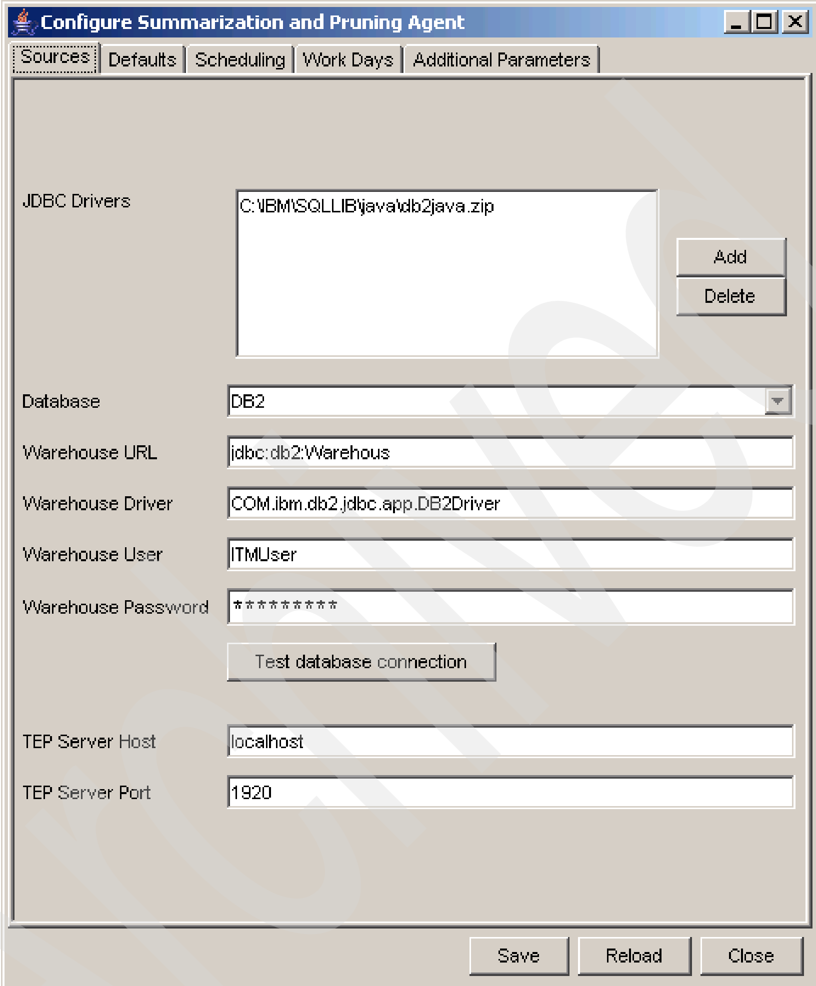


Figure 2-37 Warehouse Summarization and Pruning Agent sources configuration

7. Click the **Defaults** tab. Here, set the default settings for the Summarization and Pruning agent. By default, the summarization will be performed on the data to create monthly, weekly, daily, and hourly averages.

Note: The Pruning settings by default are to keep the yearly, quarterly, and monthly data for three years each, weekly data for two years, daily data for one year, and hourly data for three months. The default for the detailed data is to keep it for 30 days. You can change these defaults to anything that you choose and you can even override these default settings at the Aggregation Group level.

In this window, you only have to select the **Apply settings to default tables for all agents** option (Figure 2-38).

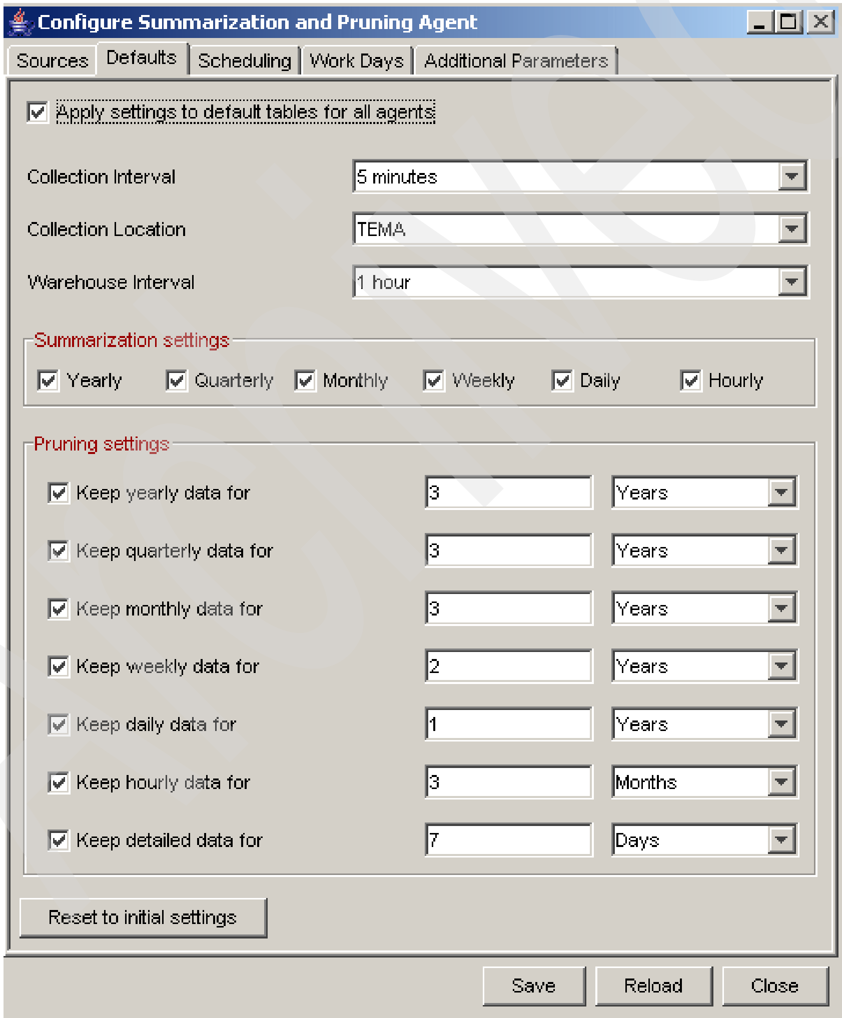


Figure 2-38 Warehouse Summarization and Pruning Agent defaults configuration

8. Form the Scheduling tab, you can specify how often the agent should run and at what times of the day. The default is every day at 2:00 a.m. We recommend that you do not change the default settings.
9. From the Work Days tab, as shown in Figure 2-39 on page 62, you can tell the Summarization and Pruning agent to differentiate between working hours and non-working hours. This feature enables you to report on your environment based on peak and off-peak hours. If you choose to use this feature, specify which hours are working hours and which are not. By default, the peak working hours are defined as any hour between 9:00 a.m. and 17:00 p.m.

   Note: If you already started collecting data in the Warehouse and you come back to make any changes to the peak and off-peak hours or vacation days, a warning message opens. It warns you that you are changing these settings and that you might have discontinuity in the data. Therefore, we suggest that you decide on your peak and off-peak hours and vacation days before you start gathering data into the Warehouse.

   You can also specify the vacation days that will be treated as off-peak days. The system differentiates automatically between weekdays and weekends. We recommend that you do not change anything.

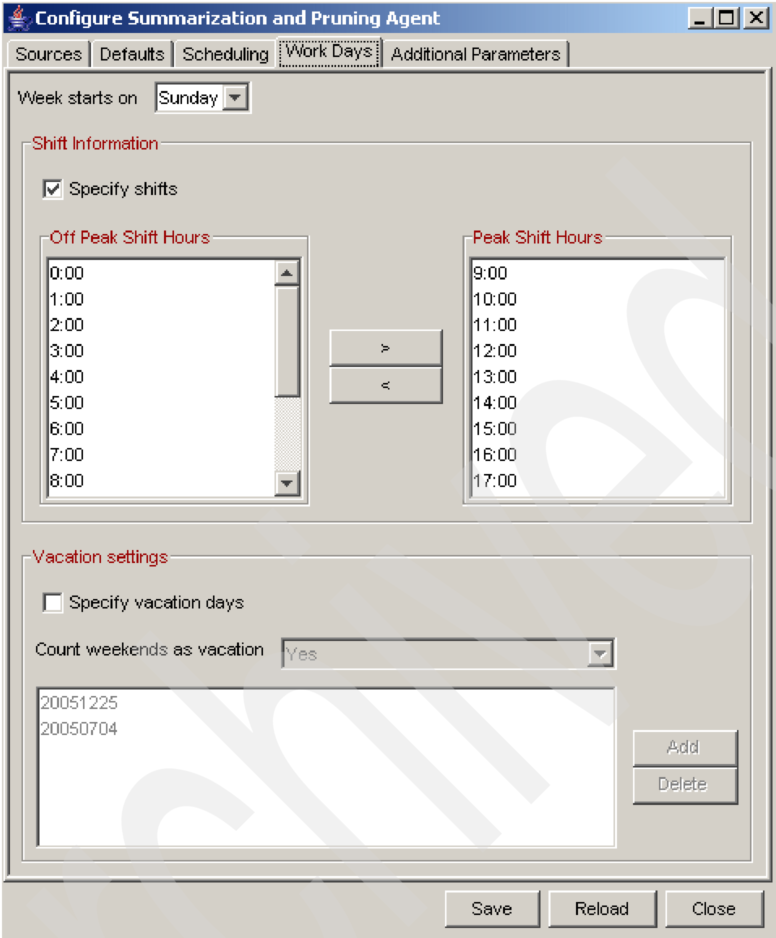


Figure 2-39 Warehouse Summarization and Pruning Agent Work Days configuration

10.From the Additional Parameters tab, you can specify how many rows per database transaction will be processed. You can use this to control the database activity, if needed, and also control after how many hours worth of hourly averages it will roll up the data to a daily average and after how many daily averages it will roll up the daily data to a weekly average. We recommend that you do not change anything.

11.Click **Save** and then **Close** to complete the configuration of the Summarization and Pruning agent.

12. In the Manage Tivoli Enterprise Monitoring Services window, you will now see that the Warehouse Summarization and Pruning agent has stopped running. Start the service again, as shown in Figure 2-40.

> **Note:** You will see that the left icon on the service side has changed to a green check symbol, but the service is stopped. When you restart the service, this icon will change to a blue runner.

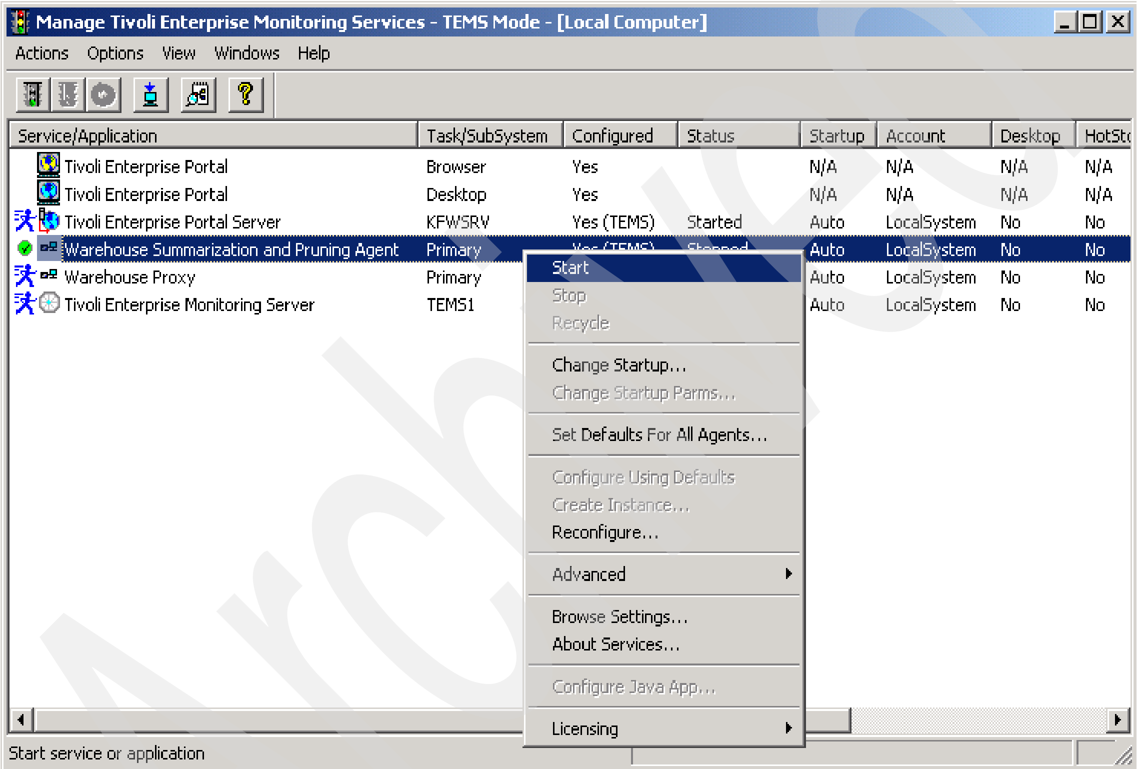


Figure 2-40 Starting Warehouse Summarization and Pruning Agent

The Warehouse Summarization and Pruning configuration is now complete.

2.5.6 Configuring hot standby Tivoli Enterprise Monitoring Server

Configure the hub and standby monitoring servers as mirrors of each other.

Installing hot standby Tivoli Enterprise Monitoring Server

Perform the following tasks to install the hot standby on a Windows computer:

1. Log in to the system with the administrator account.
2. To start the installation, go to the installation image location from IBM Tivoli Monitoring Express V6.1 CD 2. In our case, this is C:\itmexpress\disk2\Windows.
3. Click **setup.exe**. Click **Next** on the welcome window, as shown in Figure 2-41.

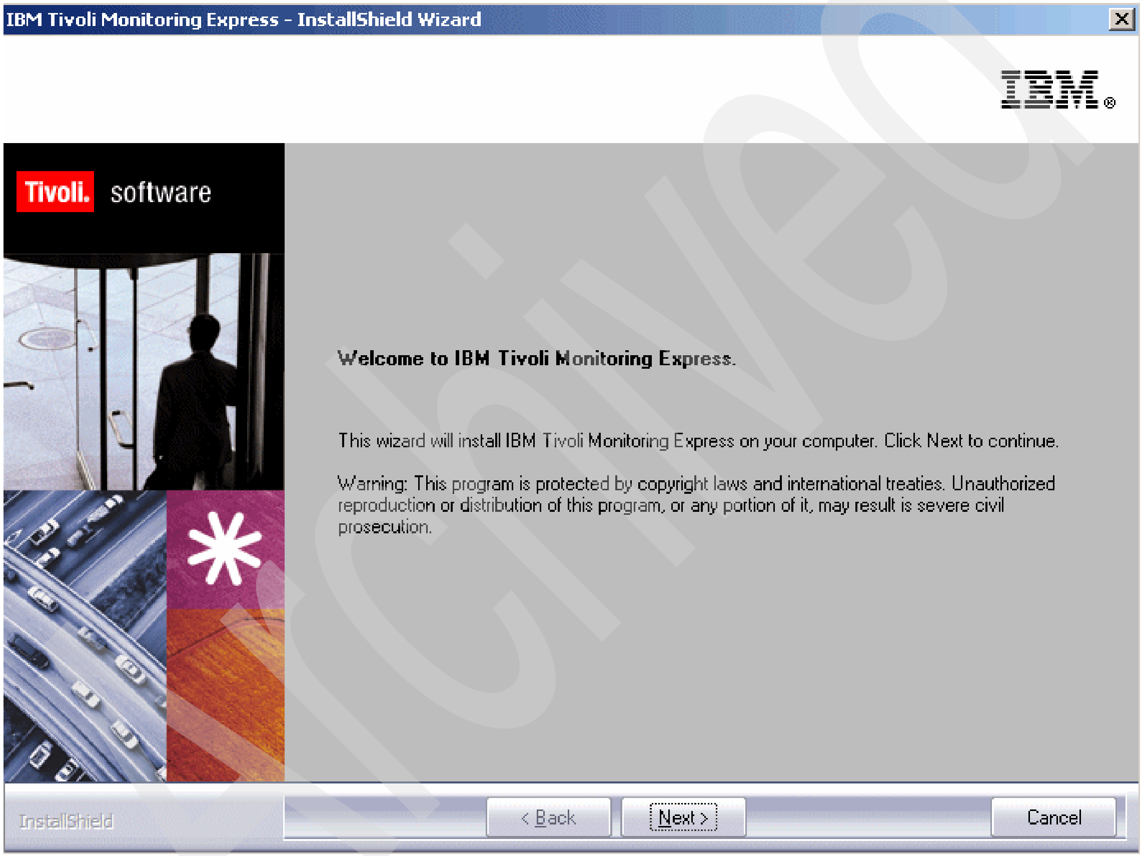


Figure 2-41 IBM Tivoli Monitoring Express welcome installation window

4. Click **Accept** to accept the license agreement.

5. If a database (DB2 or Microsoft SQL) is not installed on this computer, a message regarding potentially missing software opens, as shown in Figure 2-42. Because you do not need a database to install a Tivoli Enterprise Monitoring Server hot standby on this computer, ignore this message and click **Next**.

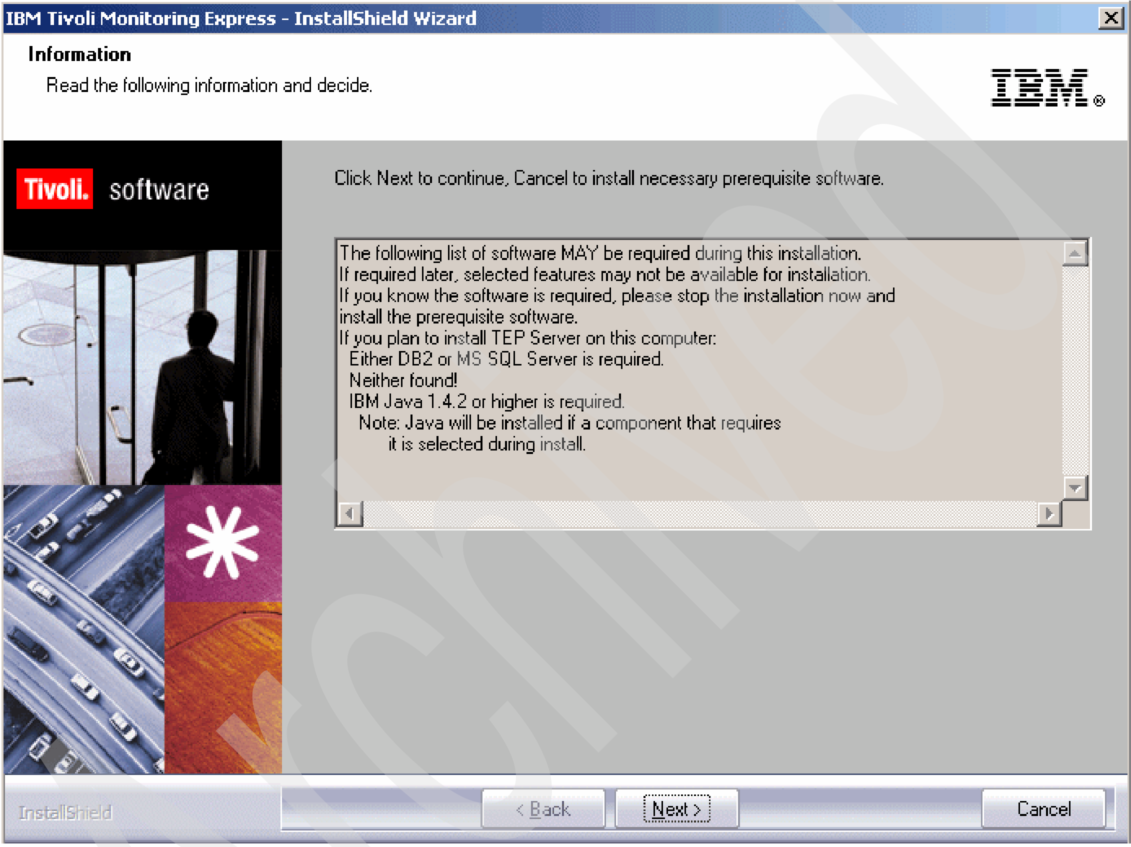


Figure 2-42 IBM Tivoli Monitoring Express requisites information

6. Choose the directory where you want to install the product. The default is c:\IBM\ITM, as shown in Figure 2-43. Click **Next**.

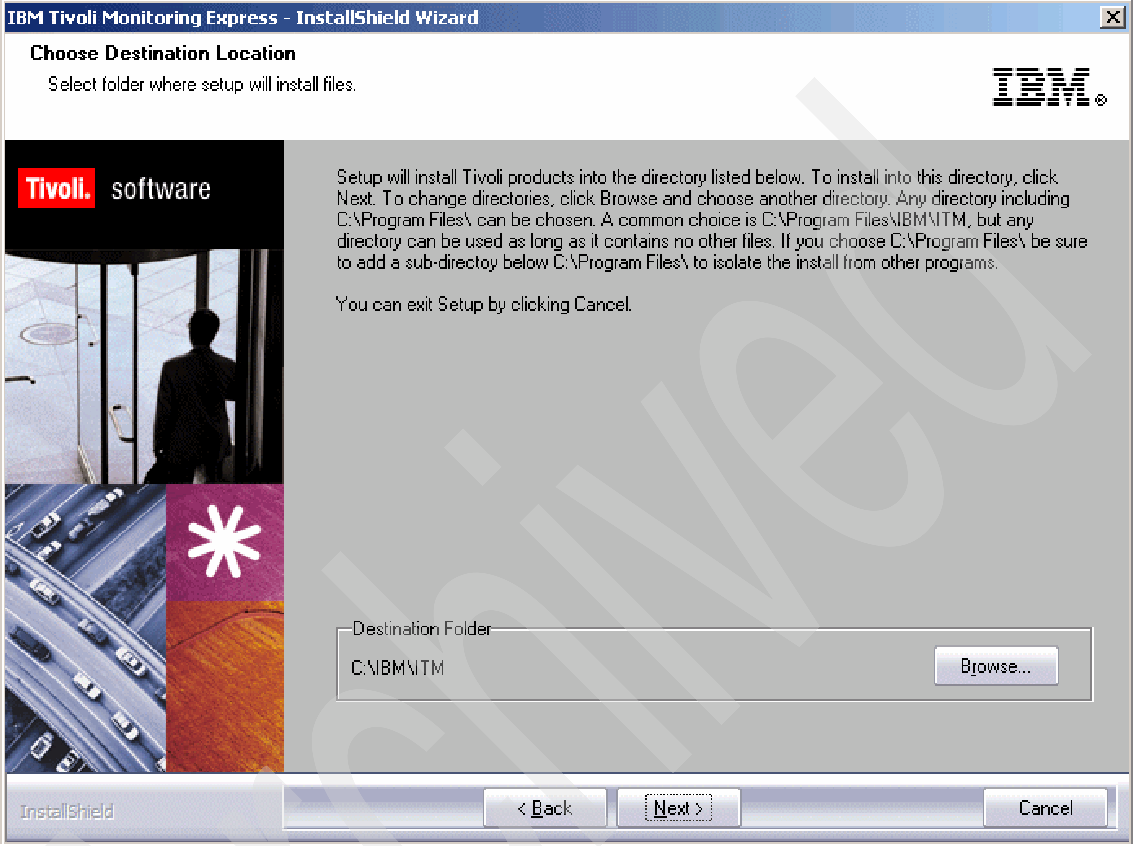


Figure 2-43 IBM Tivoli Monitoring Express default destination installation directory

7. 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, as displayed in Figure 2-44.

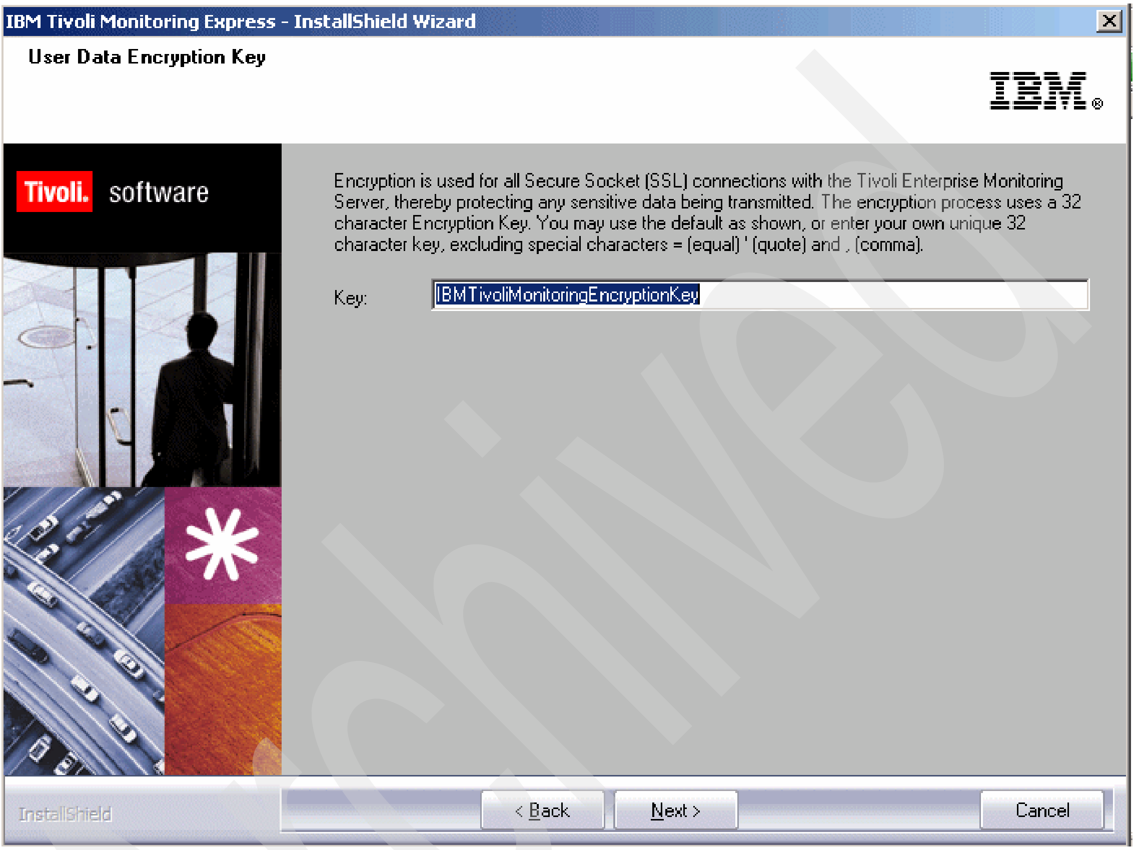


Figure 2-44 IBM Tivoli Monitoring Express encryption key confirmation

8. Expand Tivoli Enterprise Monitoring Server and select all the agents you want to install, as shown in Figure 2-45. Click **Next**.

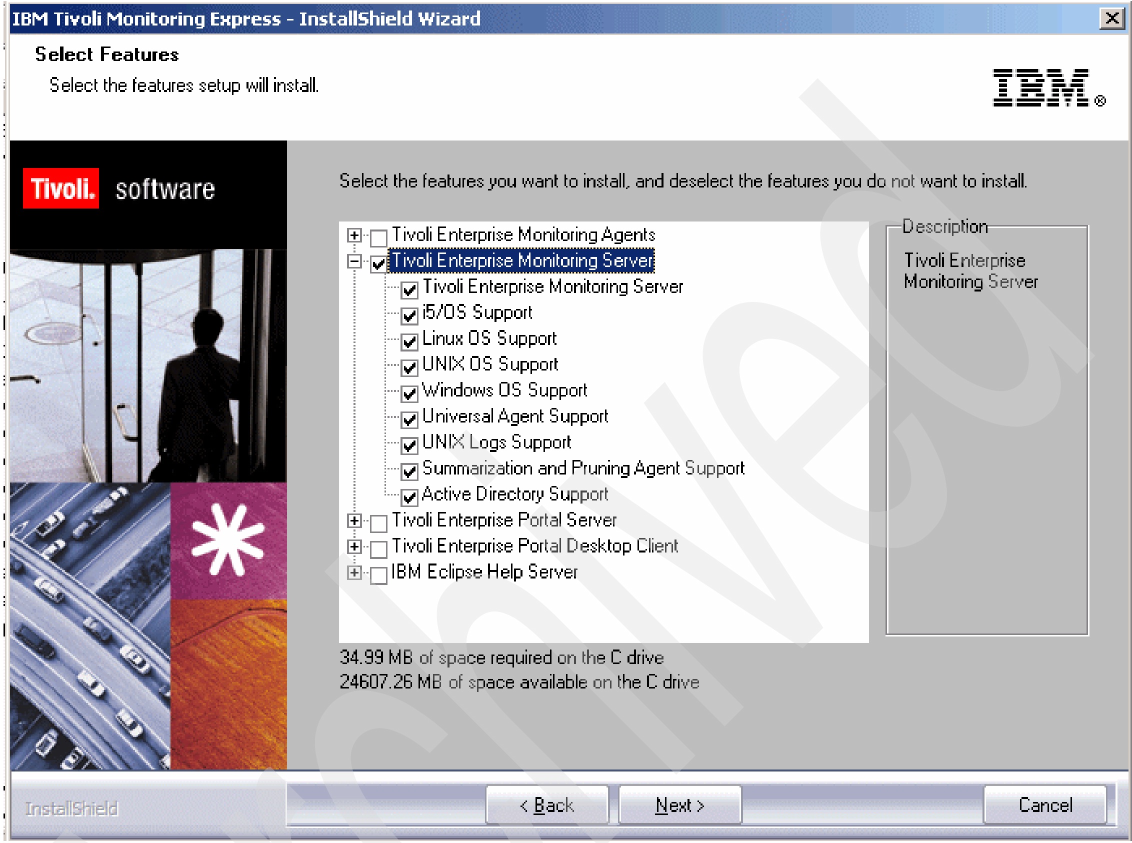


Figure 2-45 Hot standby Tivoli Enterprise Monitoring Server support to be installed

9. Click **Next** in the Agent Deploy window.

10.The IBM Tivoli Monitoring Express user ID is sysadmin. Type the required password of your choice as you did in 2.4.2, “Installing Tivoli Monitoring Express V6.1 components” on page 34. Click **Next** (Figure 2-46).

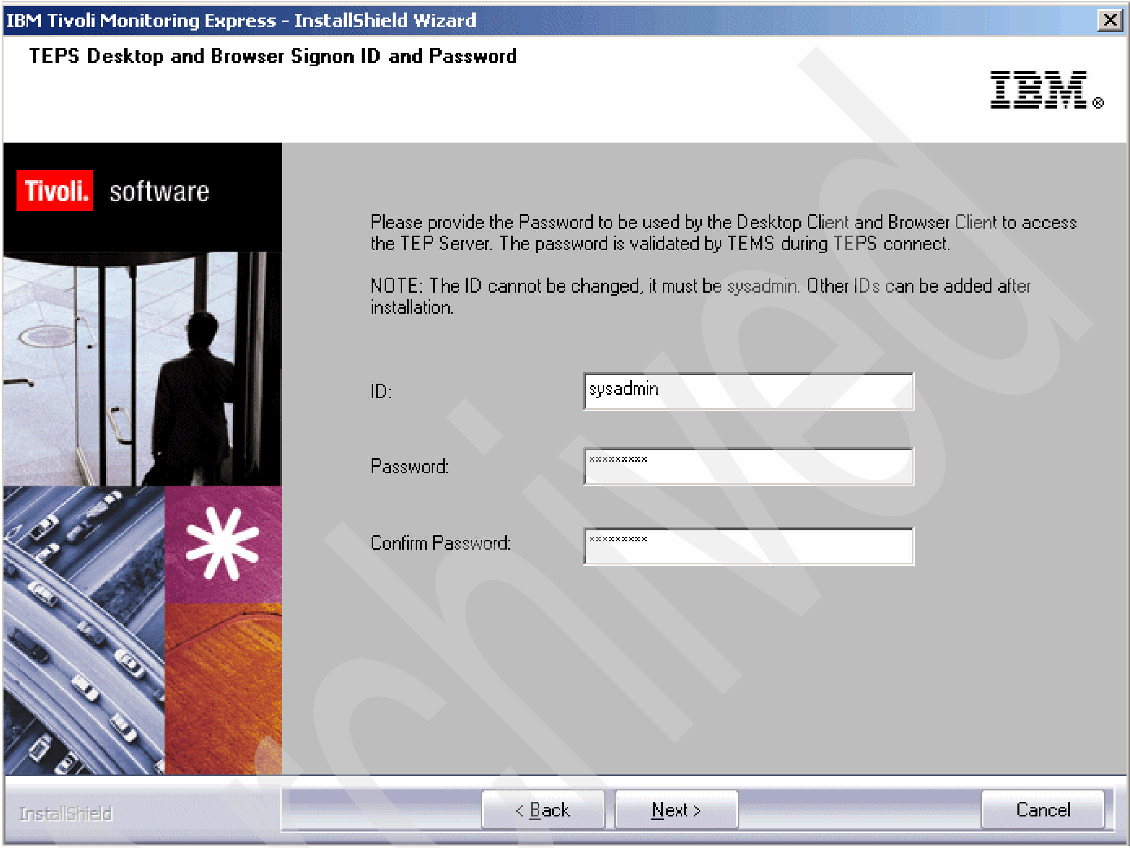


Figure 2-46 IBM Tivoli Enterprise Portal Server information

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 the hot standby Tivoli Enterprise Monitoring Server component. After the components are installed and the configuration environment is initialized, which will be indicated by a pop-up window, a configuration window opens. Click **Next**.

12. Configure the default values for your agent, as shown in Figure 2-47, and click **Next**.

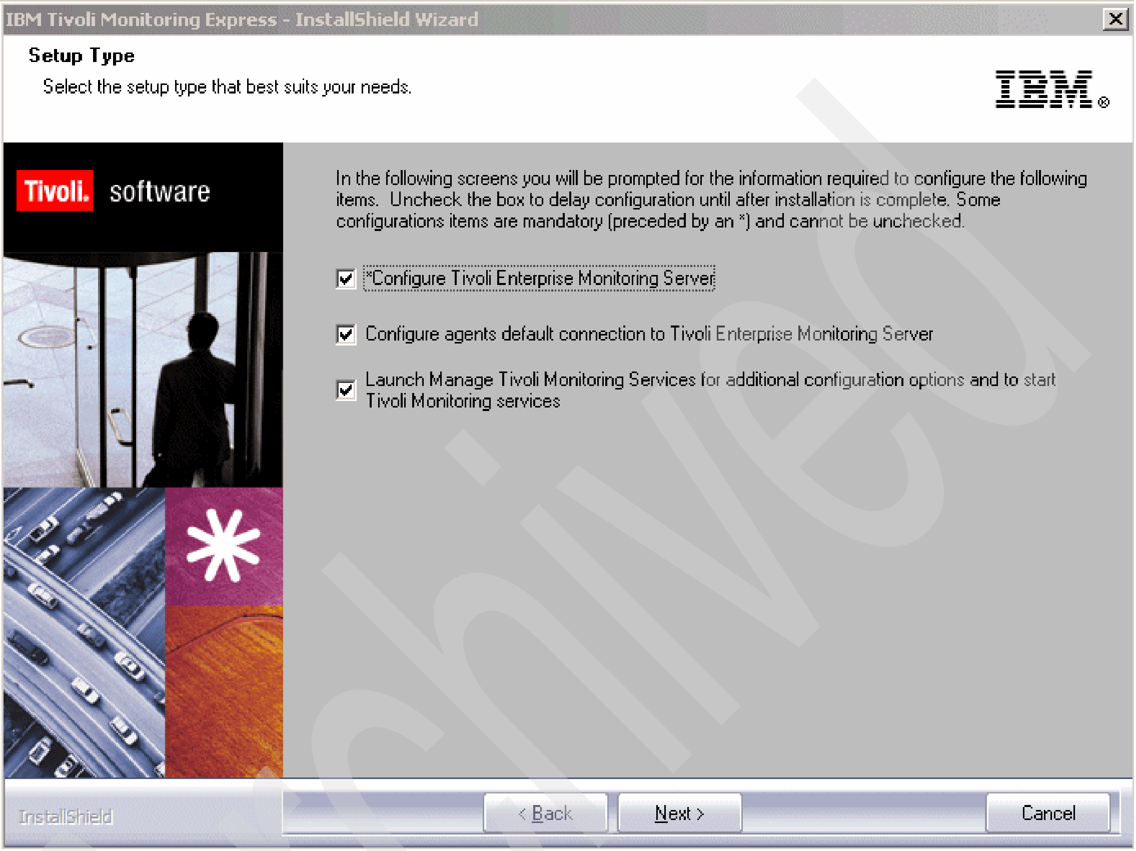


Figure 2-47 Configuration options

13. Specify the default values for any IBM Tivoli Monitoring Express agent to use when they communicate with the hot standby monitoring server:

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

   – Identify the type of protocol the agent uses to communicate with the monitoring server. Your choices are IP.UDP, IP.PIPE, IP.SPIPE, or SNA.

   – Specify three methods for communication that enables you to set up backup communication methods. If the method you have identified as Protocol 1 fails, Protocol 2 will be used.

Figure 2-48 shows the configuration of the hot standby Tivoli Enterprise Monitoring Server to which the agent will be connected. Complete the fields to define the communication between the agents and the monitoring server, and click **OK**.

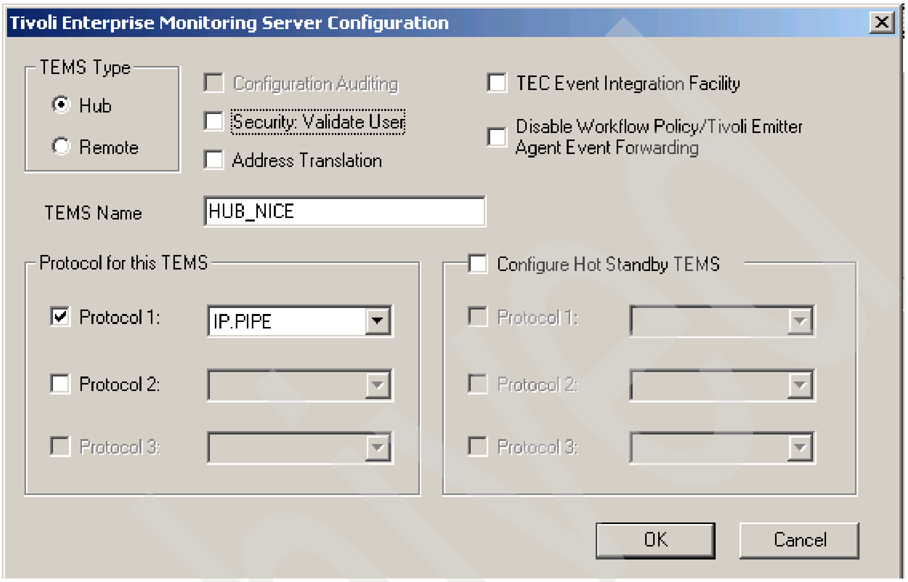


Figure 2-48 Agent communication protocols

14. Review the installation summary details, as shown in Figure 2-49, and click **OK** to complete the installation.

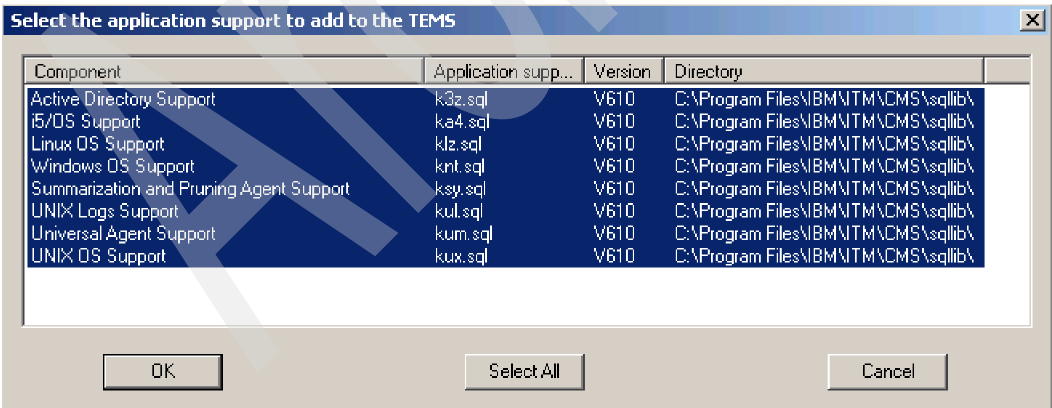


Figure 2-49 Selecting the application support

15. Open the Manage Tivoli Monitoring Services utility to see whether the Tivoli Enterprise Management Server component is configured and started, as shown in Figure 2-50. If you see `Yes` in the Configured column, it means that Tivoli Enterprise Monitoring Server is configured and started during the installation process.

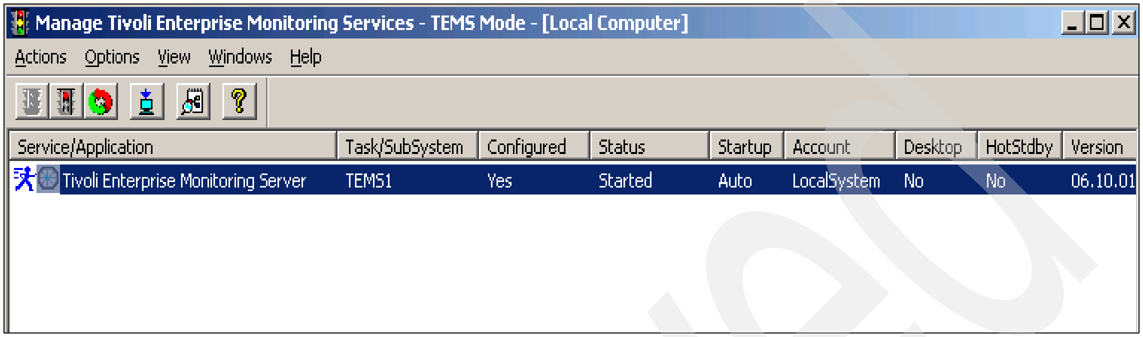


Figure 2-50 Manage Tivoli Enterprise Monitoring Services console

There is no automatic switch when the primary comes back. The hot standby expects the primary hub Tivoli Enterprise Monitoring Server and the alternate, that is, the standby hub Tivoli Enterprise Monitoring Server, to be at the same capacity. Internally, the hot standby considers them peers and does not distinguish in terms of a primary and a secondary. Instead, it handles them as acting-primary and acting-secondary hubs. Both hub Tivoli Enterprise Monitoring Servers alternate between being acting-primary and acting-secondary hubs.

By default, the algorithm that the hot standby follows to determine which is the acting-primary hub Tivoli Enterprise Monitoring Server is to query the two monitoring servers to determine how long they have been up. The hub Tivoli Enterprise Monitoring Server that has been up the longest wins.

Configuring hot standby Tivoli Enterprise Monitoring Server

To configure the hot standby Tivoli Enterprise Monitoring Server, perform the following tasks:

1. Open the Manage Tivoli Monitoring Services utility on the primary hub Tivoli Enterprise Monitoring Server to reconfigure the Tivoli Enterprise Monitoring Server component. Click **Reconfigure**.

2. Select **Configure Hot Standby TEMS** and specify the protocols used by the standby server, as shown in Figure 2-51. These protocols should match those specified for the hub server.

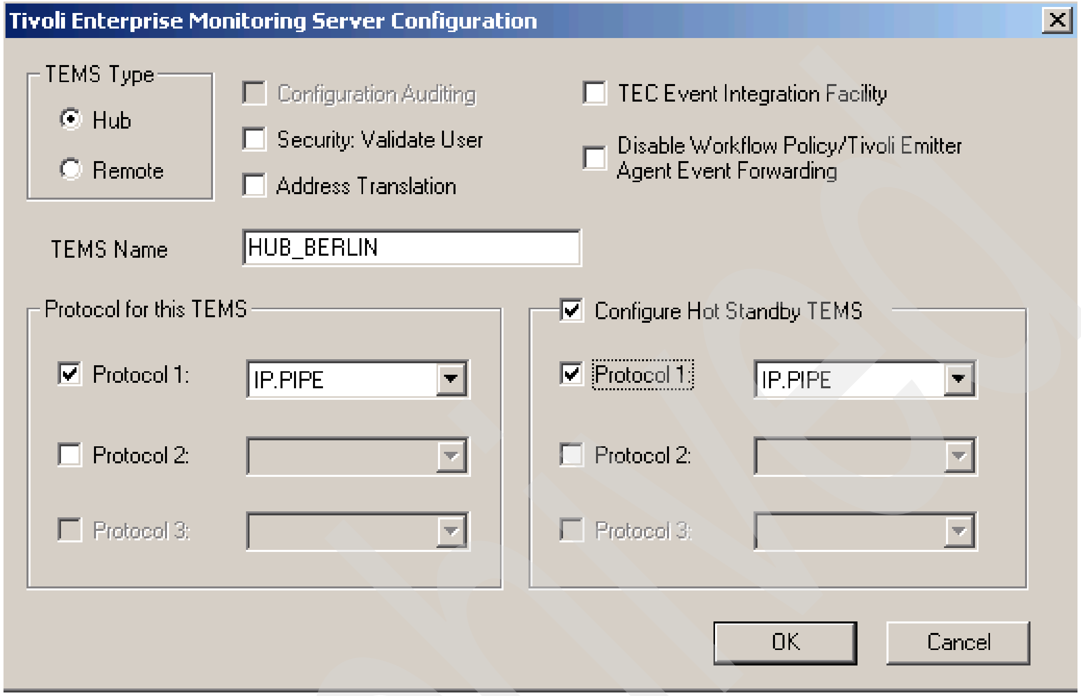


Figure 2-51 Configuring the hot standby Tivoli Enterprise Monitoring Server

3. Click **OK**. Click **OK** again on the message that opens.
4. Click **OK** in 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, as shown in Figure 2-52, and click **OK**.

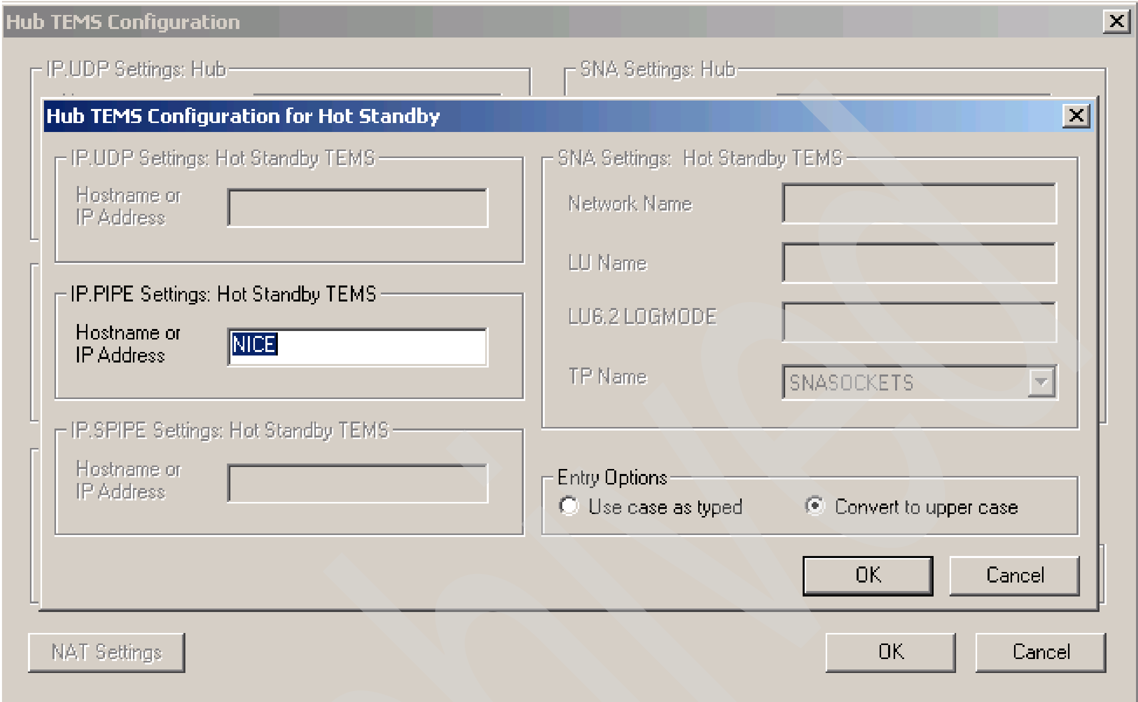


Figure 2-52 Hub TEMS Configuration for Hot Standby

6. Restart the monitoring server.

Configuring the Warehouse Proxy

The Warehouse Proxy must be configured to point to a secondary Tivoli Enterprise Monitoring Server in case the primary hub Tivoli Enterprise Monitoring Server fails and the hot standby takes its place.

To configure the Warehouse Proxy, perform the following tasks:

1. Open the **Manage Tivoli Monitoring Services** console from the Warehouse Proxy server, right-click **Warehouse Proxy** server, and click **Reconfigure**.

2. Click **OK** in the pop-up window that opens, as shown in Figure 2-53.

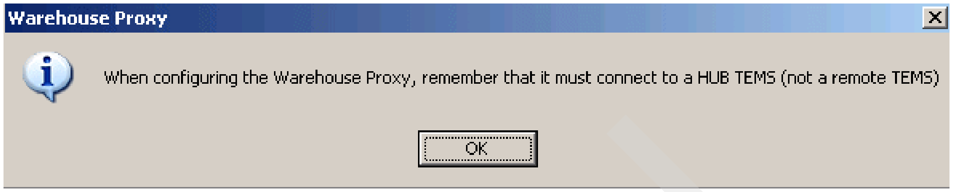


Figure 2-53 Warehouse Proxy configuration confirmation

3. The primary Tivoli Enterprise Monitoring Server communication protocol is already defined. Select **Optional Secondary TEMS Connection** and configure the communication protocol that the secondary Tivoli Enterprise Monitoring Server will use, as shown in Figure 2-54.

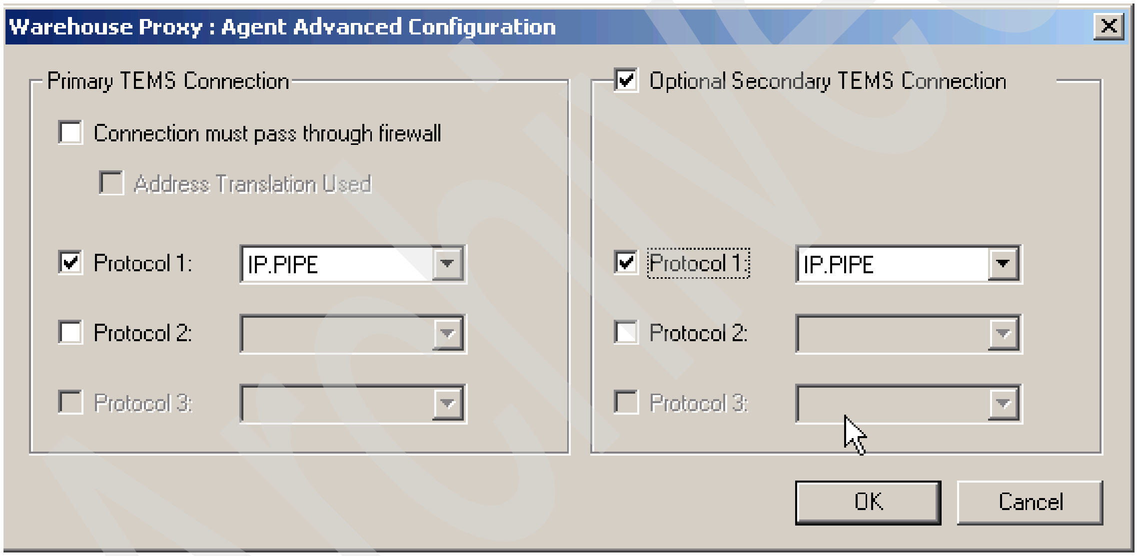


Figure 2-54 Warehouse Proxy Secondary TEMS Connection configuration

4. Click **OK** in the next window or update the primary Tivoli Enterprise Monitoring Server if it has changed, as shown in Figure 2-55.

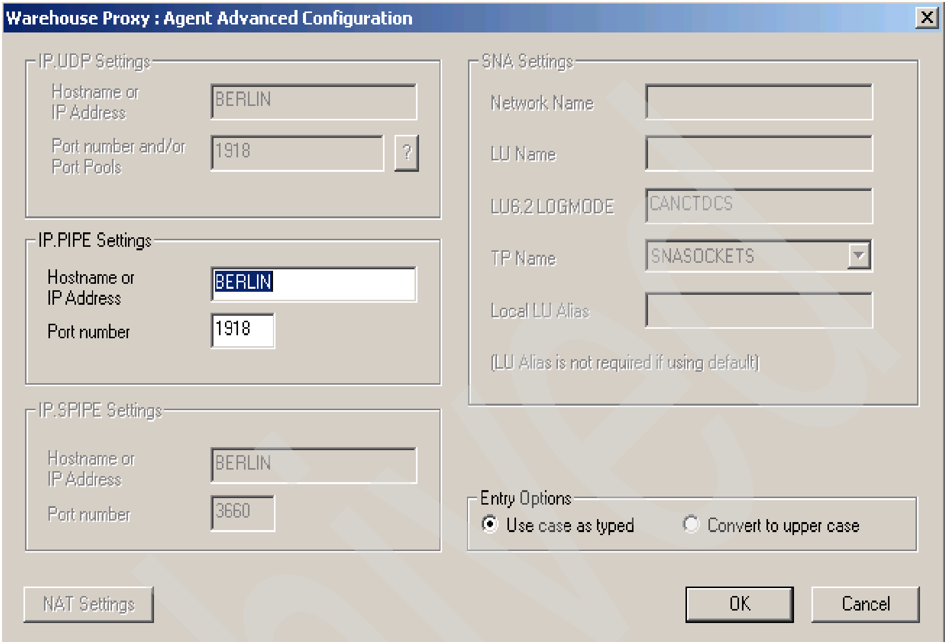


Figure 2-55 Warehouse Proxy primary Tivoli Enterprise Monitoring Server configuration

5. Type the host name or IP address for the secondary monitoring server in the Hostname or IP Address field, as shown in Figure 2-56, and click **OK**.

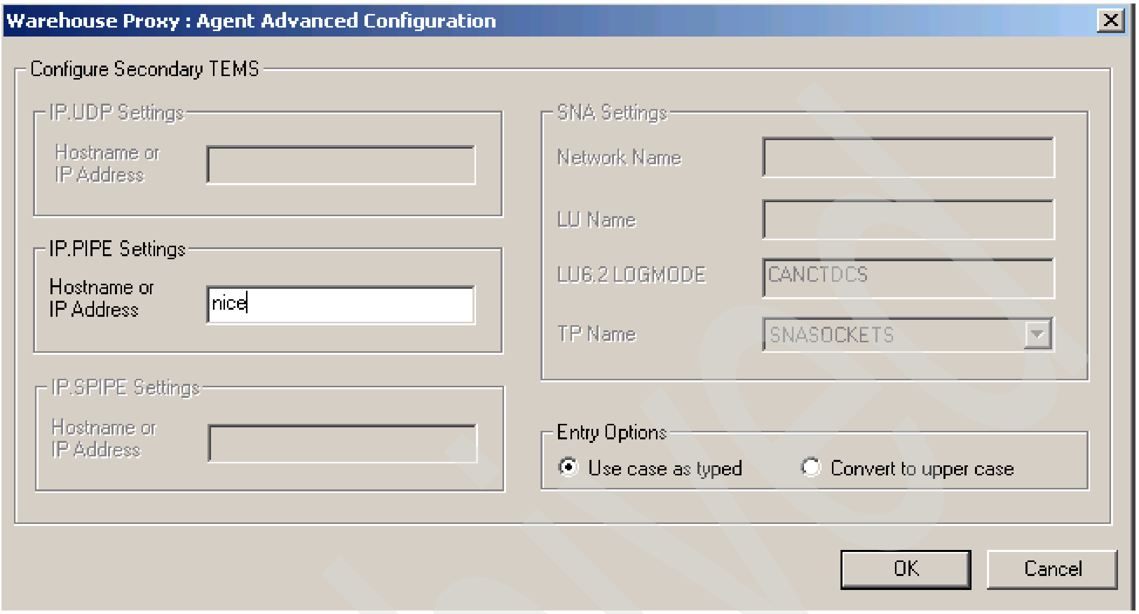


Figure 2-56 Warehouse Proxy Secondary TEMS Hostname

6. The window shown in Figure 2-57 opens. Click **No** here.

Note: You do not have to reconfigure the Warehouse Proxy.

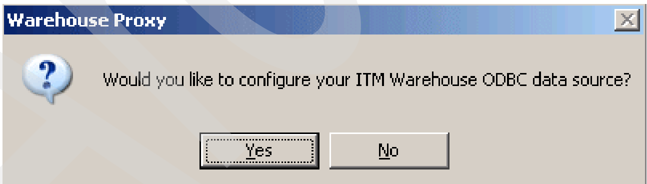


Figure 2-57 Warehouse Proxy reconfiguration

The Warehouse Proxy is now configured to connect to the secondary hub Tivoli Enterprise Monitoring Server if the primary one fails.

Start the Warehouse Proxy.

Configuring Tivoli Enterprise Portal Server

The Tivoli Enterprise Portal Server should be configured only if the primary hub Tivoli Enterprise Monitoring Server fails and the standby hub Tivoli Enterprise Monitoring Server becomes the primary.

Perform the following tasks to configure Tivoli Enterprise Portal Server:

1. Open the **Manage Tivoli Monitoring Services** console. Right-click **Tivoli Enterprise Portal Server service** and select **Reconfigure**.
2. Specify the new hub Tivoli Enterprise Monitoring Server and click **OK**.
3. The window shown in Figure 2-58 opens. Click **OK**.

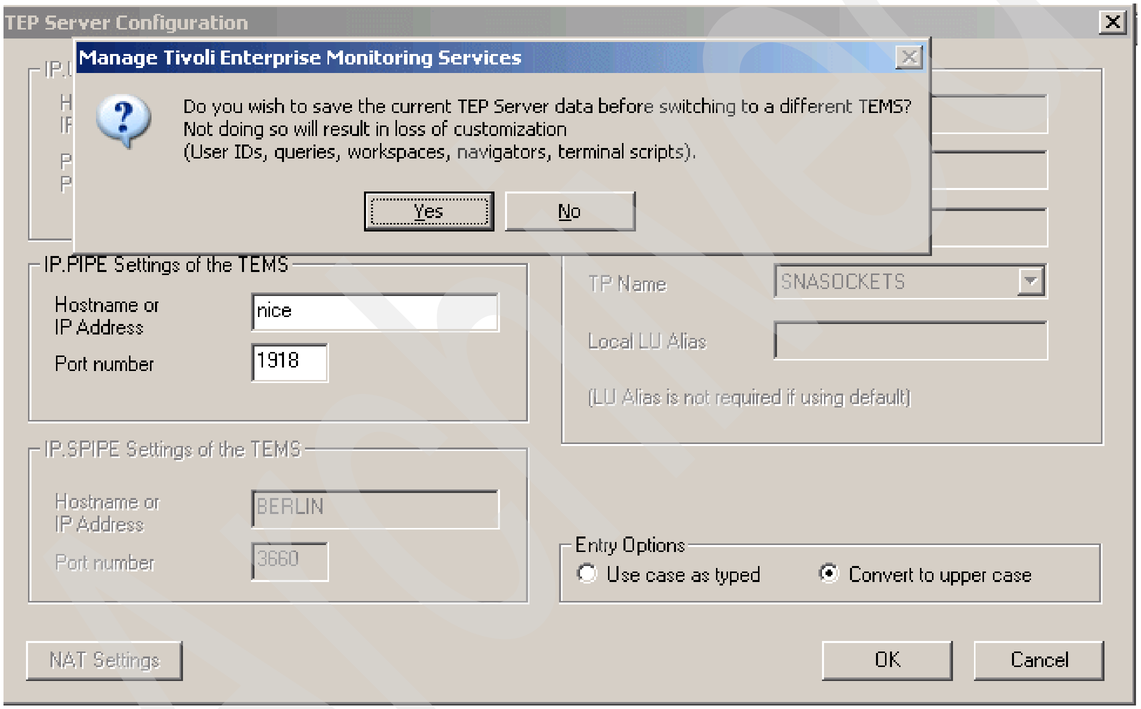


Figure 2-58   Tivoli Enterprise Portal Server configuration database backup confirmation

4. From the Enterprise Monitoring Server console, restart Tivoli Enterprise Portal Server.

> **Note:** You do not have to reconfigure the Tivoli Enterprise Portal Server database.

2.6 Installing IBM Tivoli Monitoring Express V6.1 agents

This section describes the process involved in installing and configuring different IBM Tivoli Monitoring Express V6.1 agents in a Tivoli Windows environment.

2.6.1 Deploying operating system agents

Deploy OS agents to the remote systems that you want to monitor before deploying non-OS agents to those systems. In addition to monitoring OS performance, the OS agent installs the required infrastructure for remote deployment and maintenance.

> **Note:** By default, the Windows OS agent is already configured.

Use the **`tacmd createNode`** command to deploy an OS agent from the agent depot on the server. The **`tacmd createNode`** command is so named because it creates a directory on the target computer called *Node*. This is the directory into which the OS agent is installed and non-OS agents are deployed.

As an alternative to remote deployment, you can install an OS agent locally on the target computer. Refer to *IBM Tivoli Monitoring Installation and Setup Guide,* GC32-9407, for information about how to install an agent locally.

Perform the following tasks:

1. Check if the target platform bundles are already installed in the monitoring and portal server. Execute the following command, where *hostname* is the host name where you want to initiate the installation, *user* is the user ID on the host name, and *password* is the host name password:

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

2. Deploy the agent on the targeted server by executing the following command:

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

Figure 2-59 displays the installation of the Windows agent on edinburg from berlin using the **-p** option to set the agent properties.

For the full syntax of the **tacmd createNode** command, including parameter descriptions, refer to *IBM Tivoli Monitoring Installation and Setup Guide*, GC32-9407.

```
C:\Documents and Settings\Administrator>tacmd createNode -h edinburg -d /opt/IBM
/ITM -u root
KUICCN001I Initializing required services...
KUICCN005I Enter the password for root.

KUICCN039I Attempting to connect to host edinburg ...
KUICCN050I Distributing file 45 of 45 (114.1 MB / 114.1 MB)...
KUICCN002I Beginning the installation and configuration process...

KUICCN065I The node creation operation was a success.

C:\Documents and Settings\Administrator>_
```

Figure 2-59 Deploying the agent on the targeted server

> **Note:** Unless you specifically indicate otherwise, the agent you deploy using the **tacmd createNode** command assumes that the monitoring server to which it connects is the monitoring server from which you run the command.
>
> The agent uses the default settings for the communications protocol, that is, IP.PIPE for the protocol type and 1918 for the port.

2.6.2 Deploying non-operating system agents

Deploy non-OS agents through the Tivoli Enterprise Portal or from the command line. The default non-OS agents are the UNIX logs agents, the Universal Agents, and the Active Directory agents.

The non-OS agent deployment provides the ability to update agents from a central location. It also includes carrying out a list of generic activities that must take place in order to set up a functioning monitoring agent on a machine.

This includes the following activities:

1. Determining which agents are already deployed.
2. Transferring the necessary agent bundles.
3. Installing agent bundles.
4. Configuring agent instances.
5. Starting monitoring agent instances.

Deploying through the portal

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**, as shown in Figure 2-60.

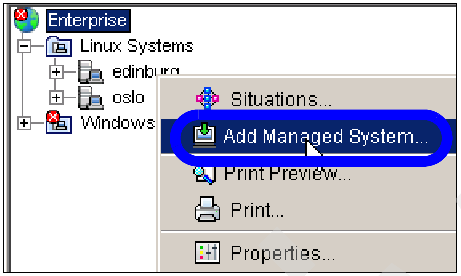


Figure 2-60 Agent deployment

4. Select the agent you want to deploy and click **OK**, as shown in the example about UNIX logs deployment in Figure 2-61.

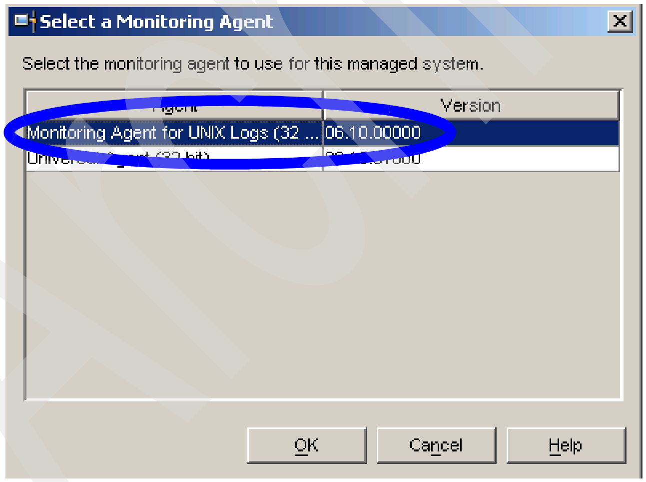


Figure 2-61 Monitoring Agent for UNIX Logs

5. Enter the information in the configuration fields required for the agent, as shown in Figure 2-62. For information about these fields, see the *User's Guide* for the agent that you are deploying. Click **Finish**.

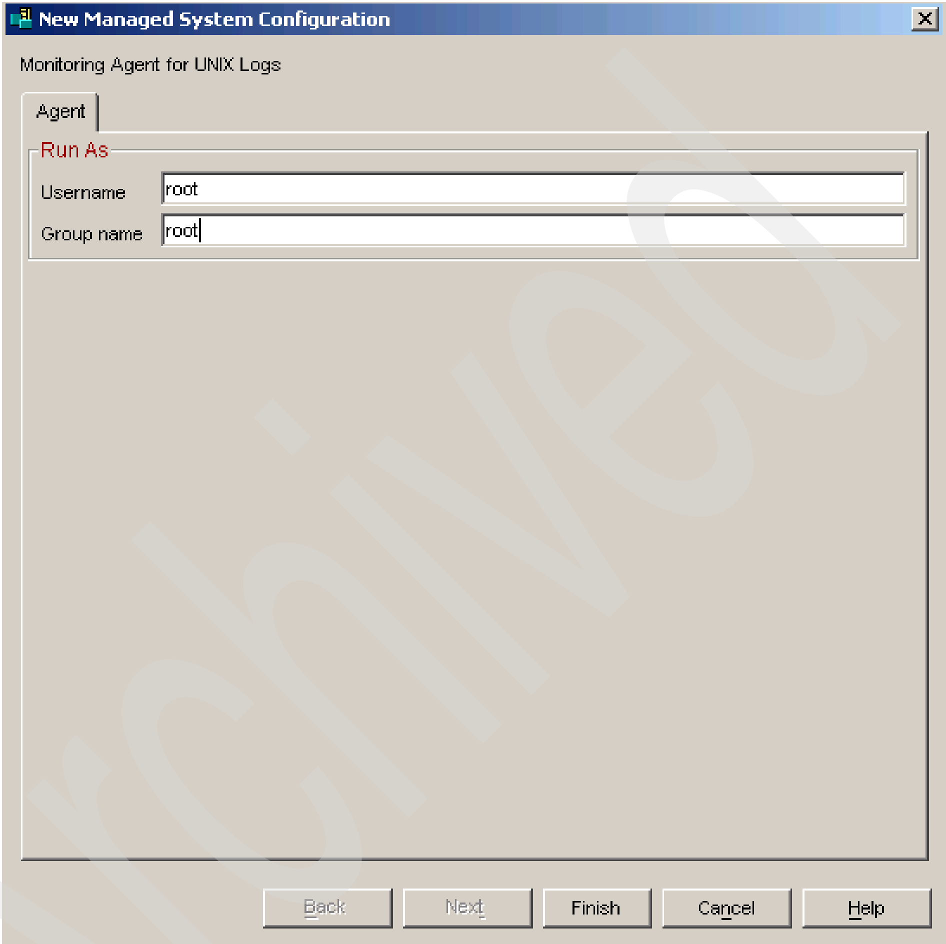
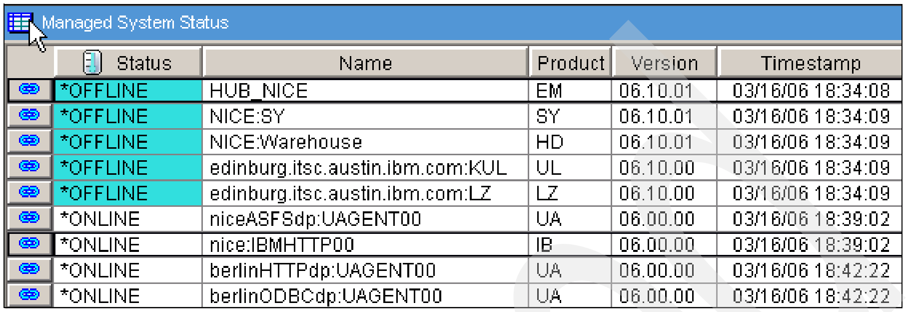


Figure 2-62 Monitoring Agent configuration fields

6. If the computer on which you are deploying the agent already has a version of that agent installed, stop the deployment and add a new instance of the agent, if possible, or reconfigure the existing agent.
7. Click **Finish** on the message window that states that the deployment was successful.

8. A view of online or offline managed systems is available from the enterprise Navigator item in the physical Navigator if you want to check the deployment. If an agent that is used for particular managed systems is not online, it is displayed in a grayed-out format, as shown in Figure 2-63.


Managed System Status

	Status	Name	Product	Version	Timestamp
	*OFFLINE	HUB_NICE	EM	06.10.01	03/16/06 18:34:08
	*OFFLINE	NICE:SY	SY	06.10.01	03/16/06 18:34:09
	*OFFLINE	NICE:Warehouse	HD	06.10.01	03/16/06 18:34:09
	*OFFLINE	edinburg.itsc.austin.ibm.com:KUL	UL	06.10.00	03/16/06 18:34:09
	*OFFLINE	edinburg.itsc.austin.ibm.com:LZ	LZ	06.10.00	03/16/06 18:34:09
	*ONLINE	niceASFSdp:UAGENT00	UA	06.00.00	03/16/06 18:39:02
	*ONLINE	nice:IBMHTTP00	IB	06.00.00	03/16/06 18:39:02
	*ONLINE	berlinHTTPdp:UAGENT00	UA	06.00.00	03/16/06 18:42:22
	*ONLINE	berlinODBCdp:UAGENT00	UA	06.00.00	03/16/06 18:42:22

Figure 2-63 Managed System Status

Deploying through the command line

You can deploy an agent using the **tacmd** command in the command line interface.

When you deploy, for example, the Universal Agent, specify an .mdl file and any scripts referenced by that .mdl file.

Before you deploy the Universal Agent, perform the following tasks:

1. Put the .mdl file in the agent depot in a UACONFIG subdirectory and the scripts in a UASCRIPT subdirectory. Create both these subdirectories under the directory C:\IBM\ITM\CMS\depot, for example.
2. Use the agent depot on the monitoring server to which the Universal Agent connects.

The following example deploys the Universal Agent (product code *um*) to the stone.ibm.com computer and specifies the UA.CONFIG property:

```
tacmd addSystem -t um -n nice.ibm.com:NT -p UA.CONFIG="IBMHTTP.mdl"
```

Each agent bundle has its own unique configuration properties that you need to provide in the **tacmd addSystem** command, using the **–p** option.

> **Note:** For the full syntax of the **tacmd** commands, including parameter descriptions, refer to *IBM Tivoli Monitoring V6.1 Installation and Setup Guide*, GC32-9407.

2.6.3 Deploying additional agents available with Tivoli Monitoring Express license

The following agent products are available:

- IBM Tivoli Monitoring for Databases
- IBM Tivoli Monitoring for Messaging and Collaboration
- IBM Tivoli OMEGAMON XE for Microsoft .NET
- IBM Tivoli Monitoring for Virtual Servers

Use the **tacmd addBundles** command to add agents that are not included on the IBM Tivoli Monitoring Express V6.1 product CDs to the agent depot. For example, run the **tacmd addBundles** command to add the agent bundles that you obtain with an IBM Tivoli Monitoring Express V6.1 license or to add maintenance packages for any agents that are already included in the agent depot.

The following procedure is an example:

1. Use the **tacmd login** command to log in to the server. The following command logs in to a server named *berlin*, with *Administrator* as the user ID and *mypassword* as the password:

   ```
   tacmd login -s berlin -u Administrator -p mypassword
   ```

2. Use the **tacmd addBundles** command to copy the new agent bundles to the agent depot. The following command copies all agent bundles from the installation media (CD image) located at C:\messaging_collaboration\WINDOWS\Deploy:

   ```
   tacmd addBundles -i “C:\messaging_collaboration\WINDOWS\Deploy”
   ```

3. View the depot to check if the new bundles are available:

   ```
   tacmd viewDepot
   ```

> **Note:** For the full syntax of the **tacmd** commands, including parameter descriptions, refer to *IBM Tivoli Monitoring V6.1 Installation and Setup Guide*, GC32-9407.

You can use both of the deployment methods to deploy the agents. For example, to deploy DB2 remotely on a Windows system through the Tivoli Enterprise Portal, follow the procedure outlined in “Deploying through the portal” on page 81 by providing the DB2 Instance Name, as shown in Figure 2-64, and the DB2 account references, as shown in Figure 2-65 on page 86.

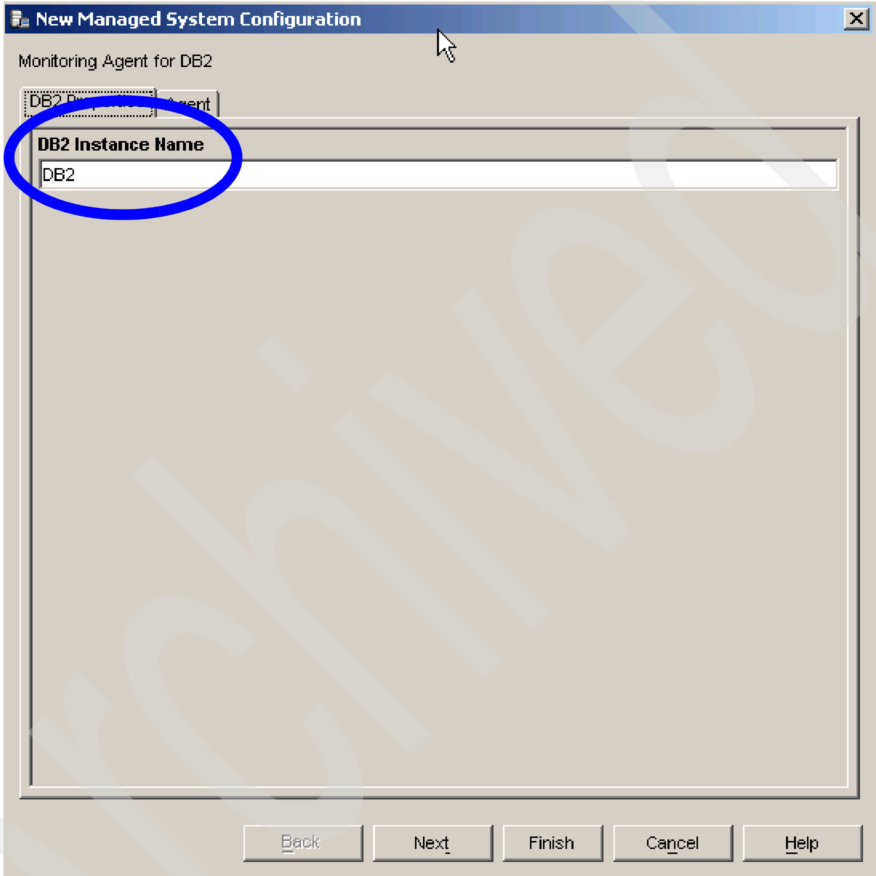


Figure 2-64 DB2 Instance Name

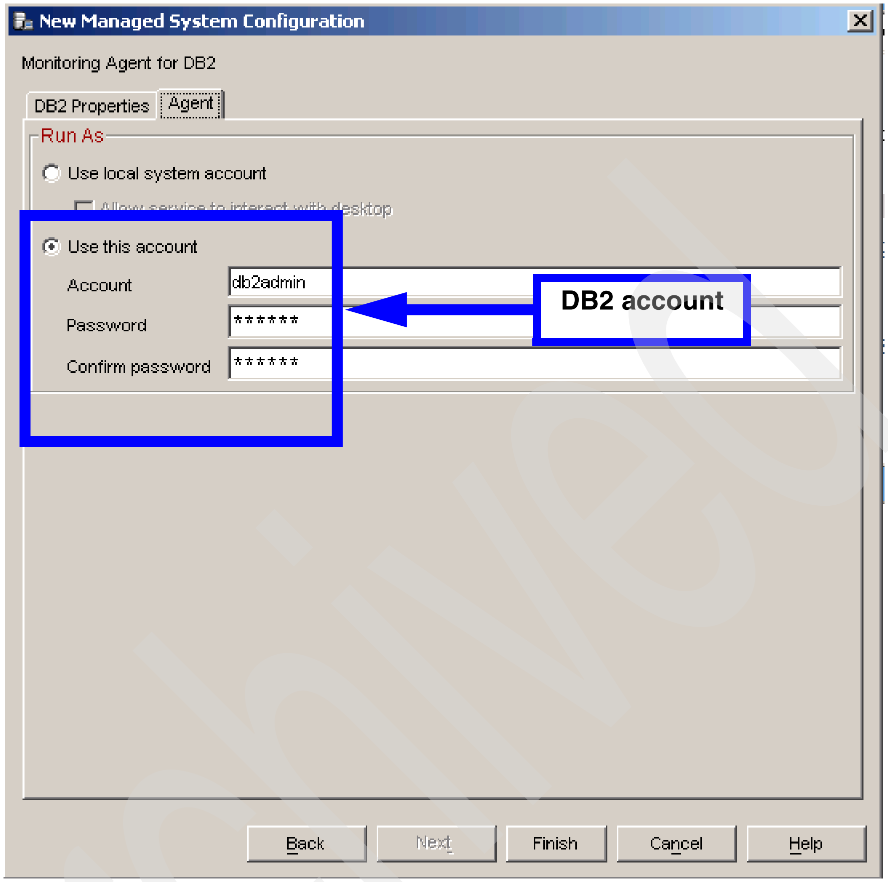


Figure 2-65 User db2admin account for deployment

You can also install the agents locally.

Perform the following tasks to install the DB2 agent:

1. Execute **`setup.exe`** in the \WINDOWS folder of the installation media.
2. This opens the Welcome window. Acknowledge this by clicking **Next**.
3. Click **Accept** to accept the license agreement.
4. You will receive an information window about the components requirements. Click **Next**.

5. A familiar window opens. Only this time, it shows the options for installing the database agents (Figure 2-66).

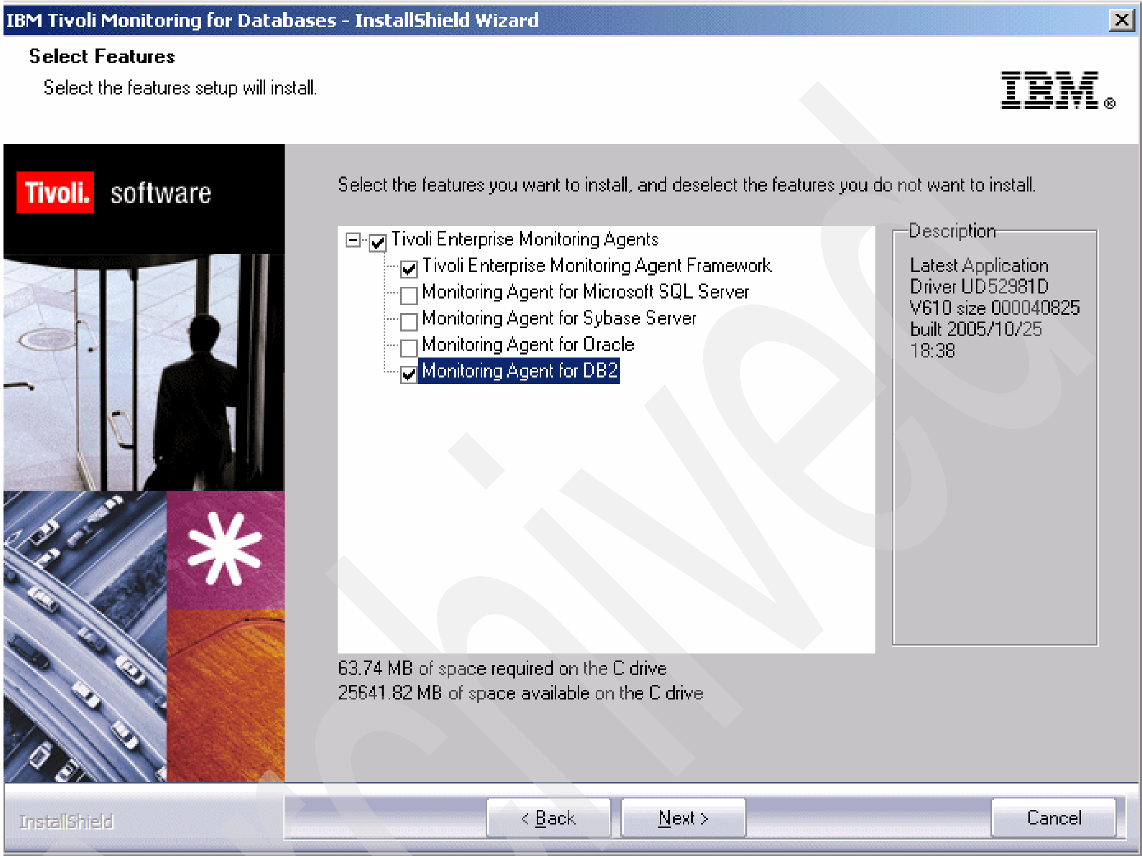


Figure 2-66 Monitoring Agent for DB2 installation

6. Expand the **Tivoli Enterprise Monitoring Agents** section and select the check box that corresponds to the DB2 agent.

   This installs the actual DB2 agent.

7. Click **Next**.

8. You will be asked which database agents to be packaged for remote distribution. If you have drive space to spare, choose the DB2 agent, or even all of them if wanted. Click **Next**.

9. The next window provides a summary of the actions that will be taken by the installation program, as shown in Figure 2-67. Click **Next** to start the installation of the DB2 agent.

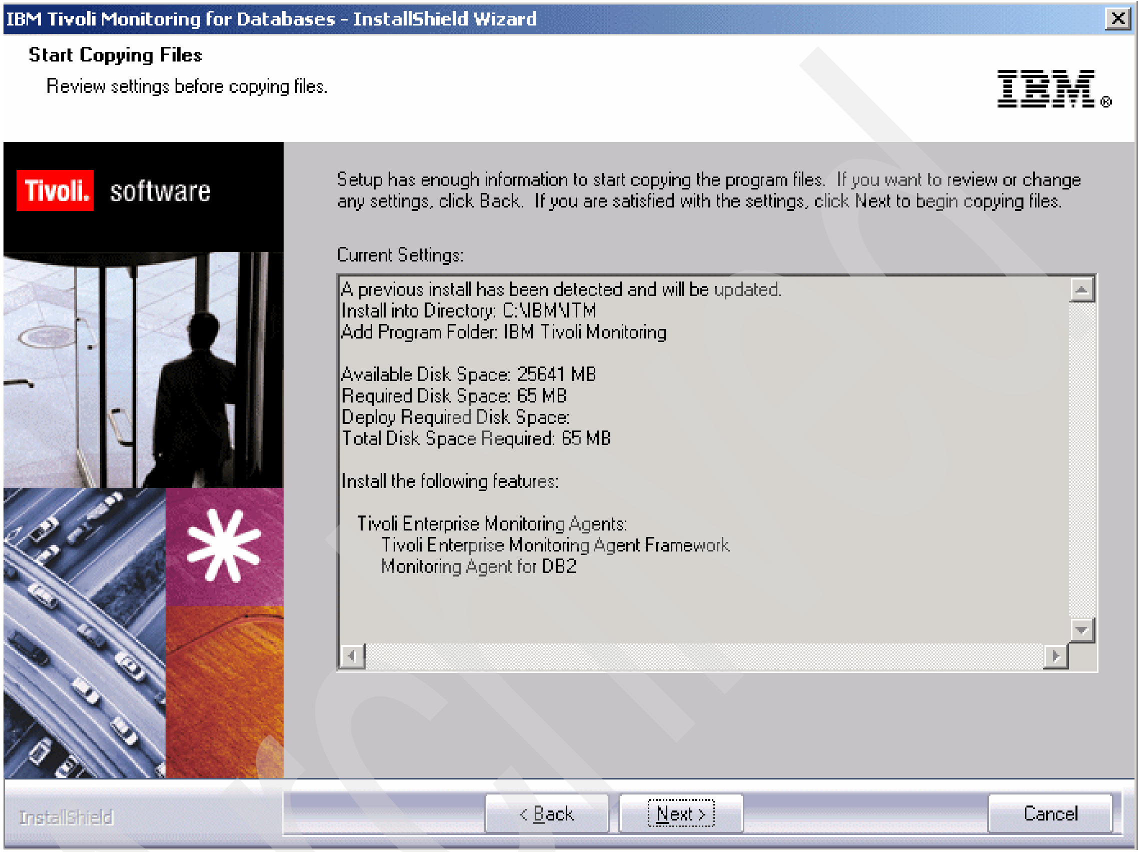


Figure 2-67 Review settings for DB2 agent installation

10. When you see a window with the stating where the Tivoli Enterprise Monitoring Server is located, as shown in Figure 2-68, accept the default by clicking **OK**.

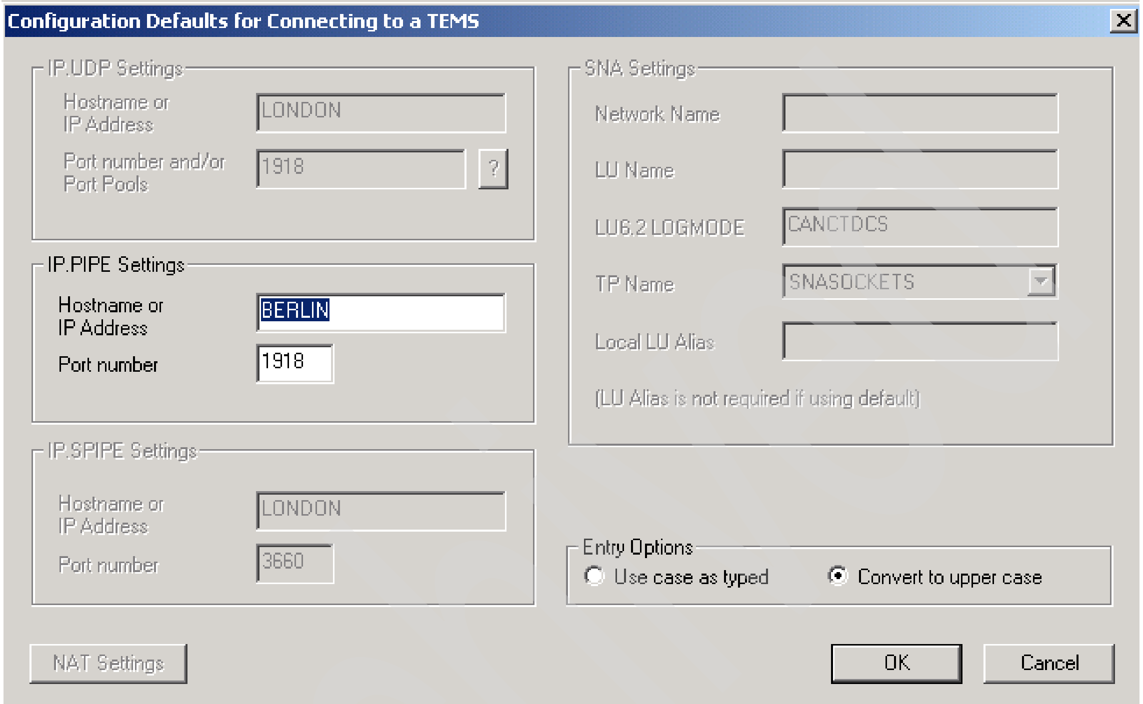


Figure 2-68 Connection to the Tivoli Enterprise Monitoring Server

11. The Manage Tivoli Enterprise Services Monitoring Services window opens. You can see that a new line, as shown in Figure 2-69. This line does not have the running person icon next to it, and in the first column, it states that this is a template.

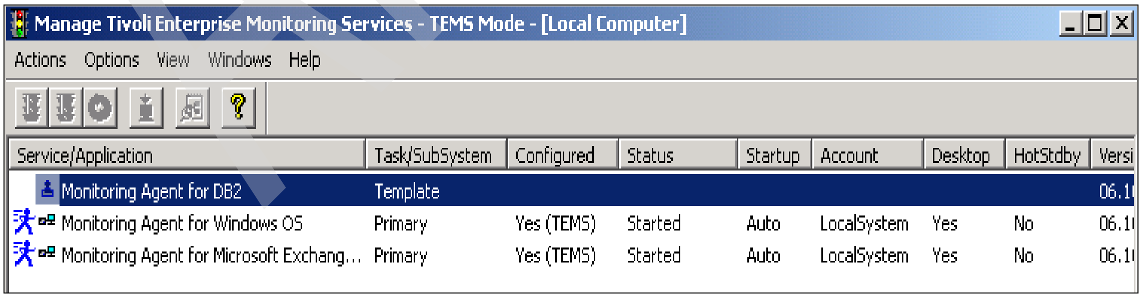


Figure 2-69 Manage Tivoli Enterprise Monitoring Services

12. Check if the support in the Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, and Tivoli Enterprise Portal client for the DB2 agent are installed on the monitoring and portal server side.

If the DB2 agent support components are not installed, install them locally in the Monitoring and Portal Server. Perform the following tasks for installation:

1. Perform from step 1 on page 86 to step 5 on page 87 of the DB2 agent installation.
2. In step 5 on page 87, expand each of the three sections and select the option corresponding to the DB2 agent, as shown in Figure 2-70. This installs the support in the Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, and Tivoli Enterprise Portal client for the DB2 agent.

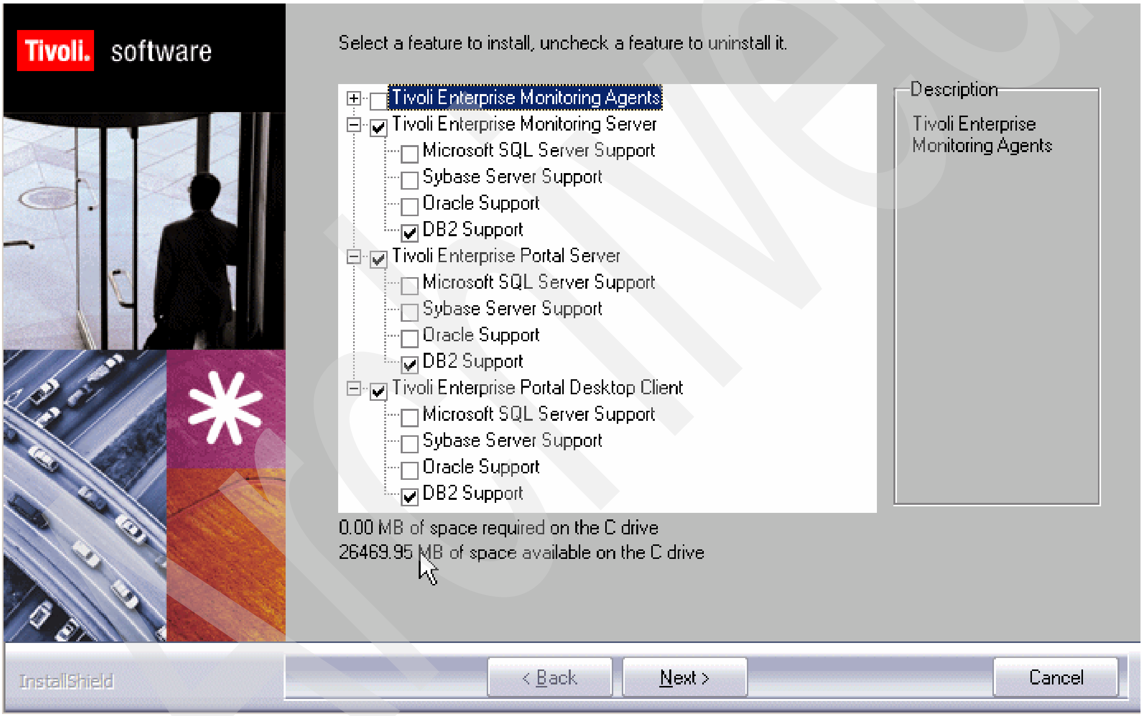


Figure 2-70 DB2 support components

3. Click **Next** until a window informs you about stopping some of the services and starts copying the files. After this is done, you will be asked about configuring the components and seeding the databases.

4. After the files finish copying, we need to configure the components. Click **Next** in the Setup Type window, as shown in Figure 2-71.

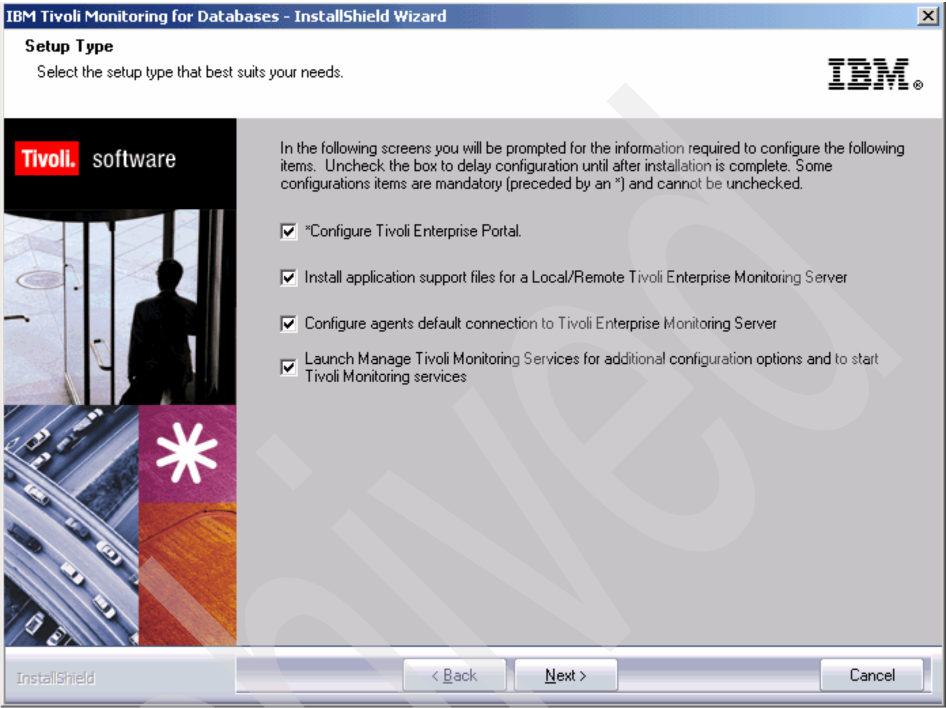


Figure 2-71 Setup Type

5. Click **Next** in the Define TEP Host Information window.
6. After a while, the Tivoli Enterprise Monitoring Server Configuration windows opens. Click **OK** to accept the Tivoli Enterprise Monitoring Server configuration, and **OK** again in Hub TEMS Configuration window.
7. Click **OK** in the Add application support to the TEMS window and **OK** again in Manage Tivoli Enterprise Monitoring Services.

8. Click **OK** in the Select the application support to add to the TEMS window. This adds DB2 support to Tivoli Enterprise Monitoring Server.

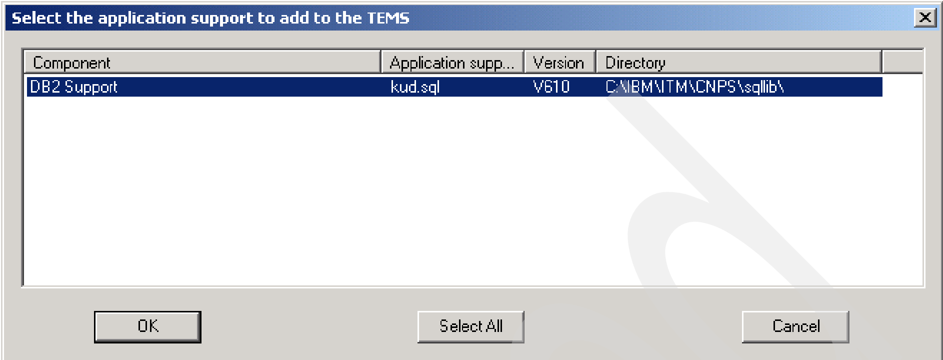


Figure 2-72 Select the application support to add to the TEMS

9. The Application support addition complete window shows installation status. `rc:0` (Figure 2-73) indicates that no error has occurred. Click **Next**.

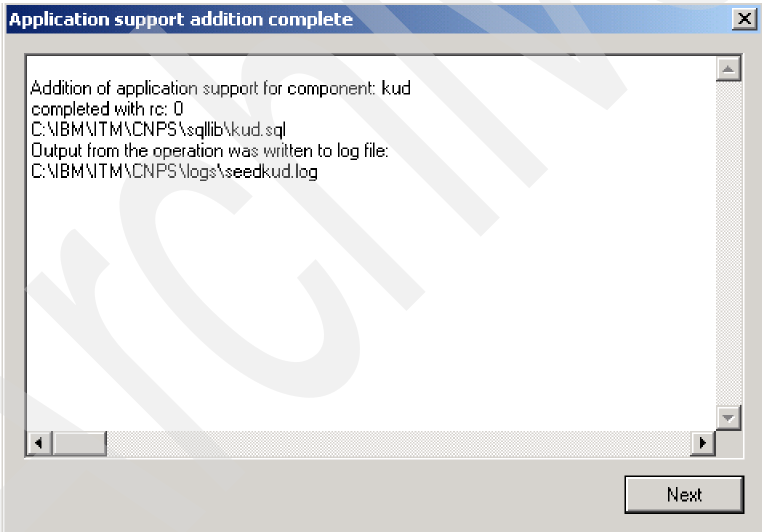


Figure 2-73 Application support addition complete

10. Select **OK** in Configuration Defaults for Connecting to a TEMS server and **OK** in Configuration Defaults for Connecting to a TEMS IP.PIPE Settings.

11. The IBM Tivoli Monitoring Services will be recycled and we can click **Finish**.

The DB2 agent must be started with an account that has administration rights in DB2. In Windows, this must be an account in the administrators group.

Configuring IBM Tivoli Monitoring Agent for Databases DB2

To configure the agent:

1. Right-click the **Monitoring Agent for DB2** and select **Configure Using Defaults**.
2. In the Monitoring Agent for DB window, type `DB2` as the DB2 instance name and click **OK**, as shown in Figure 2-74.

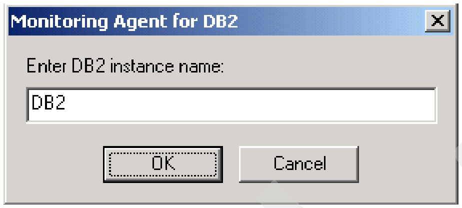


Figure 2-74 Enter DB2 instance name

3. Another service appears in Manage Tivoli Enterprise Monitoring Services, as shown in Figure 2-75.

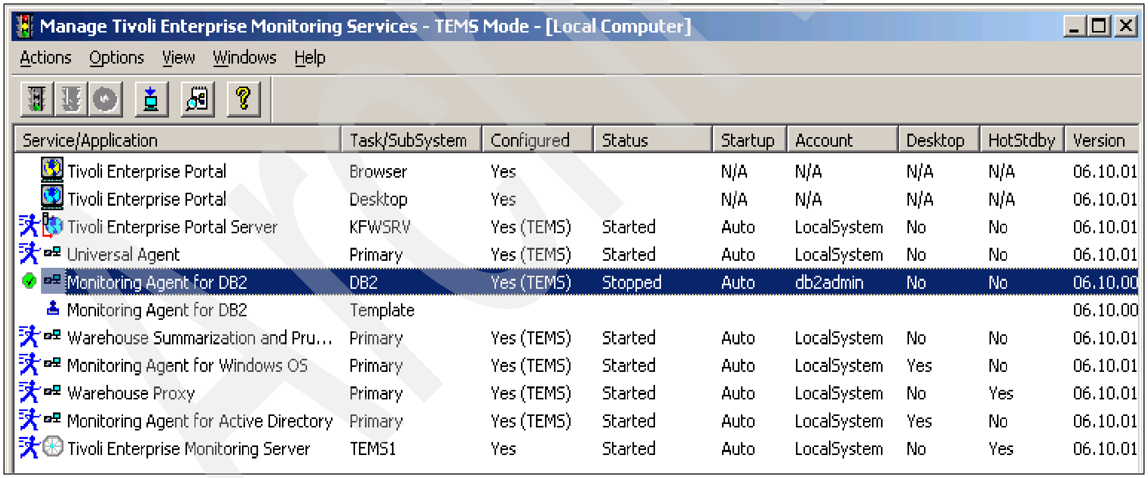


Figure 2-75 Monitoring Agent for DB2 instance

4. To start the new Monitoring Agent for DB2, right-click **Monitoring Agent for DB2** and select **Change Startup**, as shown in Figure 2-76.

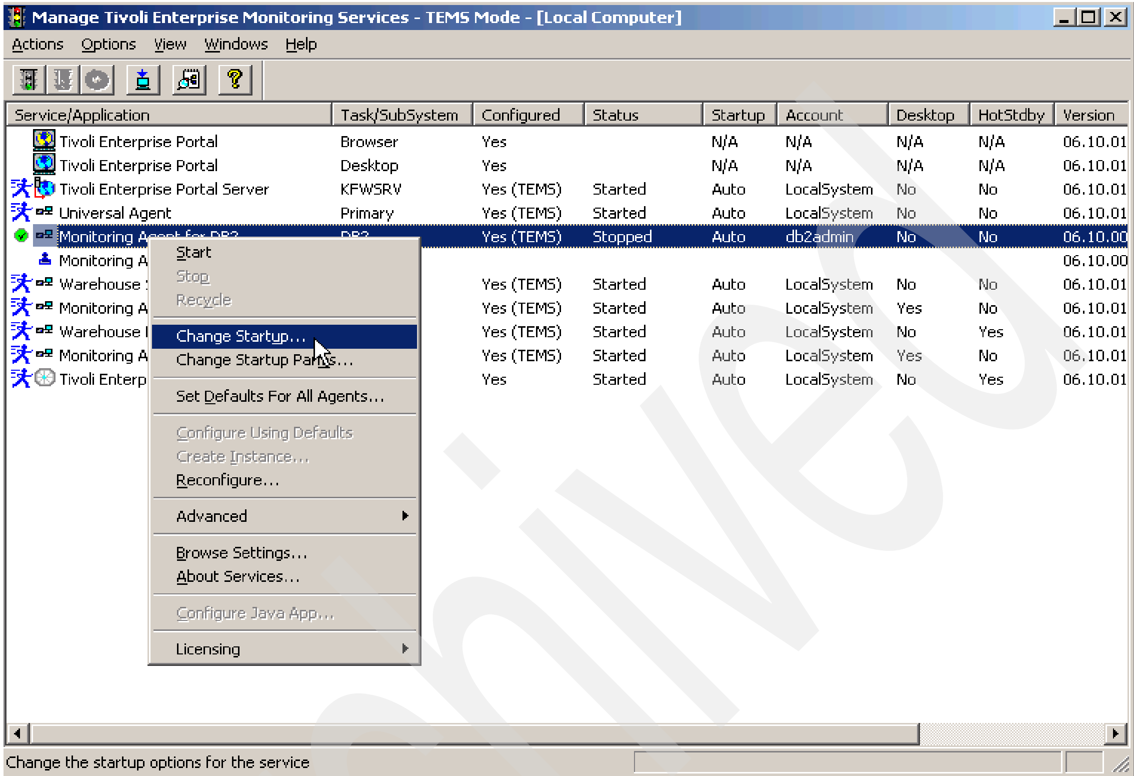


Figure 2-76 Change Startup

5. In the Service Startup for Monitoring Agent for DB2 window, click **This Account**, enter the user ID `db2admin`, with the correct password, and click **OK** as shown in Figure 2-77.

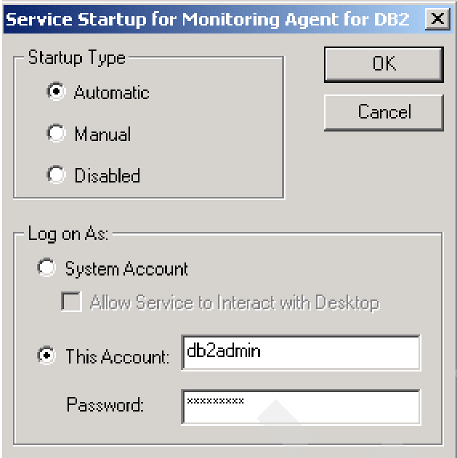


Figure 2-77 Service Startup for Monitoring Agent for DB2

6. The Service Log On Change message opens (Figure 2-78) explaining that the service will start with another account. Click **OK**.

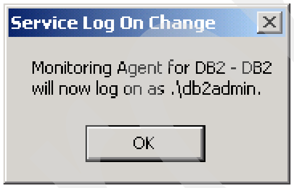


Figure 2-78 Service Log On Change

7. Double-click **Monitoring Agent for DB2** to start it.

2.7 Upgrading to IBM Tivoli Monitoring V6.1

Upgrade to IBM Tivoli Monitoring V6.1 when the following conditions occur:

- Need to monitor more than 100 servers
- Require integration with other Tivoli enterprise products
- Exceed processor limitations

When you upgrade to the enterprise version of IBM Tivoli Monitoring V6.1, you have access to the following additional features:

- The Workflow Editor, which is a feature of IBM Tivoli Monitoring. Use this to create and view policies.
- Other Tivoli enterprise products such as IBM Tivoli Enterprise Console.
- Remote monitoring servers you can use to set up a hierarchy of hub and remote monitoring servers in environments where you want to collect large amounts of distributed data.

> **Note:** For more information, refer to the IBM Redbook *Getting Started with IBM Tivoli Monitoring 6.1 on Distributed Environments*, SG24-7143.

Obtain and activate a license file for either IBM Tivoli Monitoring Express V6.1 or the enterprise version of IBM Tivoli Monitoring by following the instructions on the License Upgrade page of the IBM Tivoli Monitoring Express Launchpad:

1. Start the Launchpad by clicking the **launchpad.exe** file located in the root directory of CD1.
2. Click **License Upgrade** in the navigation pane to display the License Upgrade page.

3. Follow the instructions on the License Upgrade page, as shown in Figure 2-79. Ensure that you copy the license file of your choice in a directory that is different from the install_dir\installITM directory, where install_dir is the installation directory for IBM Tivoli Monitoring Express V6.1.

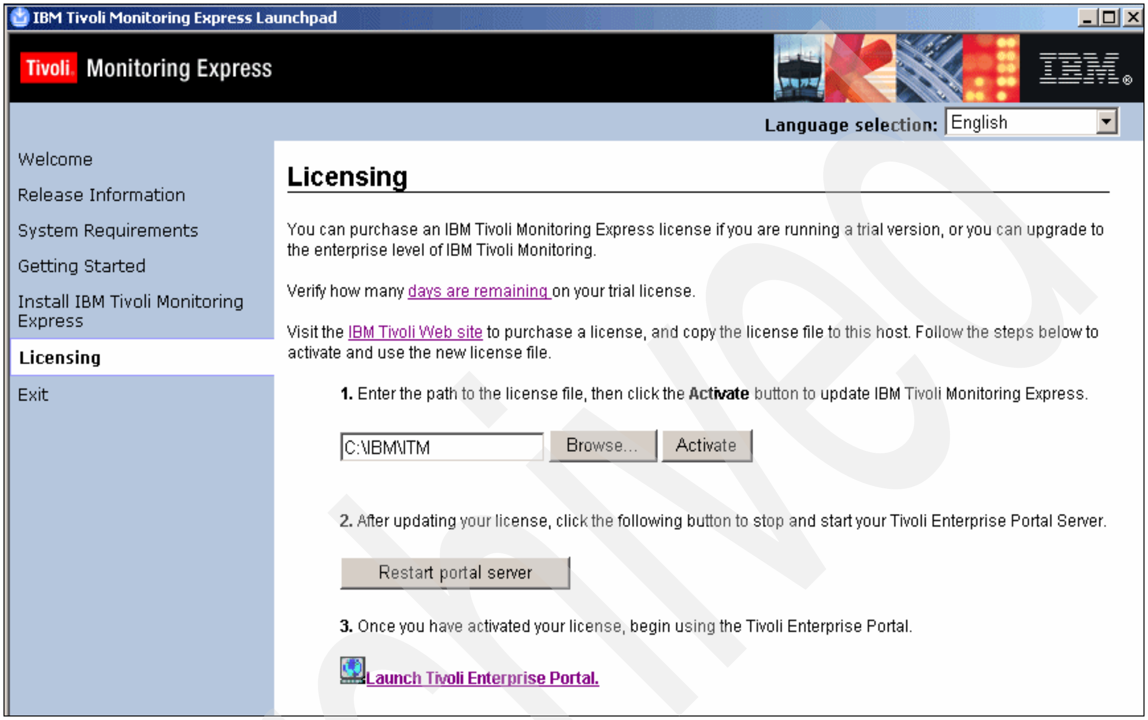


Figure 2-79 IBM Tivoli Monitoring Express License Upgrade page

4. Click **OK** in the confirmation window, as shown in Figure 2-80.

Figure 2-80 BM Tivoli Monitoring Express Launchpad confirmation

Note: You can purchase and download the license from the IBM Passport Advantage® Web site at:

http://www.ibm.com/software/sw-lotus/services/passport.nsf/%20WebDocs/Passport_Advantage_Home

2.8 Uninstalling IBM Tivoli Monitoring Express V6.1

This section describes the processes involved in uninstalling IBM Tivoli Monitoring V6.1. This includes both the entire environment and individual components.

2.8.1 Uninstalling the entire Tivoli Monitoring Express V6.1 environment

Perform the following tasks to remove the entire IBM Tivoli Monitoring Express V6.1 environment from a Windows computer:

1. From the desktop, select **Start** → **Settings** → **Control Panel** for Windows 2000, or select **Start** → **Control Panel** for Windows 2003.
2. Click **Add/Remove Programs**.
3. Select **IBM Tivoli Monitoring Express** and click **Change/Remove**.
4. Select **Remove** and click **Next**.
5. Click **OK**.
6. After Tivoli Enterprise services has stopped, you will be 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, opens.
8. Select **Yes** to restart your computer and click **Finish**.

2.8.2 Uninstalling an individual agent or component

Perform the following tasks to remove a component on a Windows computer:

1. Log on to the system with the administrator account.
2. To start the installation, go to the installation image location from the IBM Tivoli Monitoring Express V6.1 CD 2. In our case, this is C:\itmexpress\disk2\Windows.
3. Click **setup.exe**.

4. Perform one of the following tasks:
   - To uninstall a specific agent or component, select **Modify**.
   - To uninstall the entire agent bundle, select **Remove**.
5. Click **Next**.
6. Perform one of the following tasks:
   - If you are uninstalling an agent bundle, click **OK** to confirm the uninstallation.
   - If you are uninstalling an agent or component, perform the following tasks:
     i. For an agent, expand **Tivoli Enterprise Monitoring Agents** and select the agent you want to uninstall. For a component, select the component, for example, **Tivoli Enterprise Portal Desktop Client**.
     ii. Click **Next**.
     iii. Click **Next** in the confirmation window.
     iv. Depending on the remaining components on your computer, there might be a series of configuration panels. Click **Next** in each of them.
7. Click **Finish**.
8. Restart the computer to complete the uninstallation.

> **Note:** When removing a specific component (Modify/Remove), do not clear any component other than the one you are removing. Clearing any other component will uninstall it from the machine.

3

Historical summarized data

This chapter describes the architecture, planning, and implementation of IBM Tivoli Monitoring Express V6.1 historical data collection. One of the primary features of the new IBM Tivoli Monitoring Express V6.1 product is the historical database. In this chapter, we discuss the overall architecture of how historical data is collected on IBM Tivoli Monitoring Express V6.1 agents, how historical data is collected by a IBM Tivoli Monitoring Express V6.1 warehouse server (also referred as Tivoli Data Warehouse V2.1 or Tivoli Data Warehouse), and how historical summarization and pruning occurs within the historical database. We explain the details of the architecture of IBM Tivoli Monitoring Express V6.1 historical data collection and how the historical data can be accessed.

This chapter also uses real-world scenarios to plan, design, and configure IBM Tivoli Monitoring Express V6.1 historical data. The scenarios include the configuration of the historical data for the following agents: Windows OS monitoring, Linux OS monitoring, and DB2 on UNIX monitoring. We also discuss some of the reporting tools that can be used with the historical database.

We discuss the following topics in this chapter:

- Overview
- Architecture
- Planning: Logical configuration considerations
- Configuring
- Reporting: Accessing IBM Tivoli Data Warehouse from Tivoli Enterprise Portal

3.1 Overview

The IBM Tivoli Monitoring Express V6.1 historical database has two processes that collect, summarize, and prune data gathered from IBM Tivoli Monitoring Express V6.1 agents:

- A Warehouse Proxy agent is the historical database server. The Warehouse Proxy agent collects data from the IBM Tivoli Monitoring Express V6.1 agents and stores the data in a relational database (IBM DB2 Express or Microsoft SQL).
- The historical database can optionally be configured to summarize and prune the historical data with another new process called the Summarization and Pruning agent.

Figure 3-1 illustrates an overview of historical data collection.

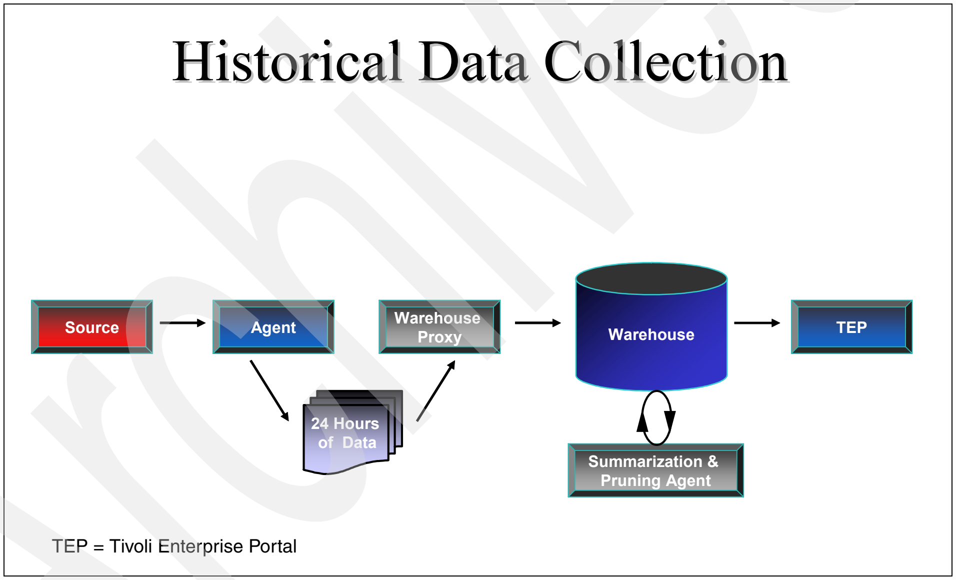


Figure 3-1 Historical data collection overview

We discuss these new processes in more detail in the following section.

3.2 Architecture

In this section, we discuss the architecture of IBM Tivoli Monitoring Express V6.1 historical database. The main topics that are discussed in this section are:

- Historical data architecture overview
- Historical data types
- Component flows
- Data tables and attributes

3.2.1 Historical data architecture overview

The IBM Tivoli Monitoring Express V6.1 historical data collection architecture consists of three primary components. The following components are used to collect data in the IBM Tivoli Monitoring Express architecture:

- Warehouse Proxy agent
- Summarization and Pruning agent
- Historical database (data warehouse)

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.

Summarization and Pruning agent

The Summarization and Pruning agent maintains the data within the historical database by aggregation and pruning data based on customer specifications. The Tivoli Monitoring Express V6.1 administrator sets up how often to collect the detailed data, what intervals on which aggregate and prune, and how often to run the aggregation and pruning engine. Typically, the summarization and running process is scheduled to run once a day.

Historical database

The Tivoli Monitoring Express V6.1 historical 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 Tivoli Enterprise Portal if historical data is to be presented. External reporting tools and other applications can access the data and operate off of this database.

3.2.2 Historical data types

There are two types of data stores for the Tivoli Monitoring Express historical database:

- Short-term data
- Long-term data

Short-term data

Short-term data is typically referred to in Tivoli Monitoring Express V6.1 as data that is stored in binary files and is less than 24 hours old. In the Tivoli Monitoring Express V6.1 architecture, short-term data can be configured to store the binary files locally on the Tivoli Enterprise Monitoring Agent or it can be configured to store the binary files on the Tivoli Enterprise Monitoring Server. This can be configured by a user by agent type. In both cases (Tivoli Enterprise Monitoring Agent or Tivoli Enterprise Monitoring Server) 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 historical database by the Warehouse Proxy agent, it is pruned on the Tivoli Enterprise Monitoring Agent or Tivoli Enterprise Monitoring Server if it is older than 24 hours. If the Warehouse Proxy agent is not configured to collect the short-term data, a user-defined pruning job must be implemented. We recommend that the binary short-term data be located on the Tivoli Enterprise Monitoring Agent (that is, the agent). We discuss this configuration option in 3.4, “Configuring” on page 128. The binary short-term data will never be in aggregate or summarized format regardless of whether it is stored on the Tivoli Enterprise Monitoring Agent or the Tivoli Enterprise Monitoring Server.

Long-term data

Long-term data in Tivoli Monitoring Express V6.1 is typically referred to as data that is older than 24 hours and has been collected up to the historical database to the Warehouse Proxy agent. The long-term data resides in tables in the historical database. The long-term RDBMS tables contain detailed data and summarized data in the database. The Summarization and Pruning agent can be configured to run every day to roll up data from the detailed level to hourly, weekly, monthly, quarterly, or yearly. The Summarization and Pruning agent also prunes the summarized tables (Figure 3-2 on page 105).

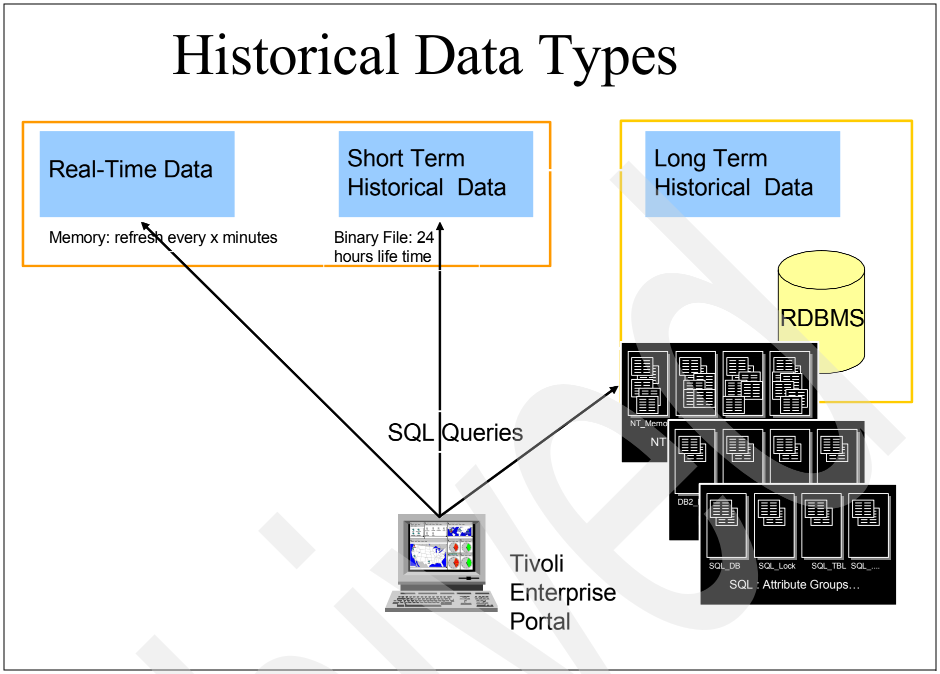


Figure 3-2 Historical data types

3.2.3 Component flows

When historical data collection is configured in Tivoli Monitoring Express V6.1, a user can determine whether the short-term data (the binary 24-hour data) should be stored on the Tivoli Enterprise Monitoring Server or on the Tivoli Enterprise Monitoring Agent. If the data is stored on the Tivoli Enterprise Monitoring Agents, each monitored machine stores binary files for all of the monitoring agents running on that system. In some cases, it might be necessary to collect short-term historical data on the Tivoli Enterprise Monitoring Server (that is, the server). 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 monitoring server, the binary files for all monitored machines and their agents will be collected up to the monitoring server. This creates 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 configured, data is loaded from the Tivoli Enterprise Monitoring Agent or Tivoli Enterprise Monitoring Server (depending on the location setting) to the historical database RDBMS. When data is collected to the Warehouse Proxy agent, tables are created in the historical 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 historical database tables, data older than 24 hours will be pruned from the short-term binary files located on the monitoring agent or monitoring server. At any given time, you can have 24 hours of short-term raw data on the monitoring agent or monitoring server and detailed tables in the historical database RDBMS that contains the same data. When a request is made from the Tivoli Enterprise Portal to perform a query that uses the time span function, data is retrieved from the binary file if the time span is less than or equal to 24 hours. A query performed from the Tivoli Enterprise Portal that uses a time span greater than 24 hours will retrieve data from the historical database 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 historical database. The Tivoli Enterprise Monitoring Server determines where to get the data: either from the agent, if the data is less than 24 hours old, or from the historical database, 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 3-3 shows the flow of historical data collected when the location is stored on the Tivoli Enterprise Monitoring Agent.

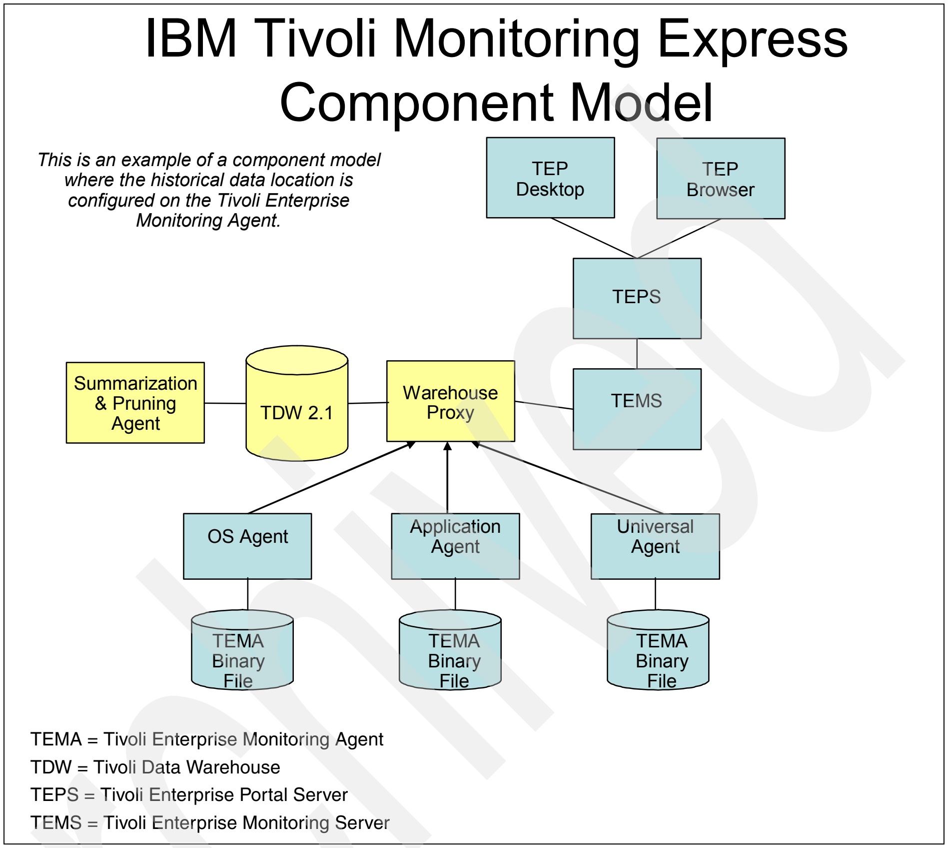


Figure 3-3 Historical collection location Tivoli Enterprise Monitoring Agent

3.2.4 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 Express V6.1 agent. For example, the Tivoli Monitoring Express 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 Express V6.1 agent that describes the agent's attribute groups and attributes.

Table 3-1 is an example of three IBM Tivoli Monitoring Express V6.1 agents and their default attribute groups.

Table 3-1 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 Tivoli Monitoring Express V6.1, the default attribute groups can be configured to collect historical data (see 3.4, “Configuring” on page 128). After the data collection starts, the agent starts storing short-term binary tables on the Tivoli Enterprise Monitoring Agent or Tivoli Enterprise Monitoring Server depending on the collection location that has been configured. For example, Table 3-2 on page 109 lists four agents’ default binary file table names. These are the names of the binary file tables as they appear on the monitoring agent or monitoring server.

Table 3-2 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

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 time stamp of the HDR file can be useful to determine the first time that data collection took place for that attribute group. The time stamp on the table name (that is, the file without the *.hdr) indicates the last time data collection occurred for that attribute group. Using the time stamps 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 Tivoli Enterprise Portal 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 can 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 are in the default IBM Tivoli Monitoring Express V6.1 installation 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 historical database is a single RDBMS database. The supported databases are DB2 V8.2 or later and Microsoft 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 Monitoring Express V6.1 historical database: one detailed table and multiple summarization tables for each attribute group. For example, if yearly, quarterly, monthly, weekly, daily, and hourly summarization is turned on for the NT_Memory attribute group, the following tables are created in the 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 historical database table names are created with quotation marks surrounding the table name. When referencing historical data in the database, you must use double quotation marks to ensure correct access to that data.

Figure 3-1 on page 114 shows how to use an SQL query to get a list of all table names. Figure 3-4 is a list of the NT_Memory tables in the historical database from the DB2 V8.2 Control Center.

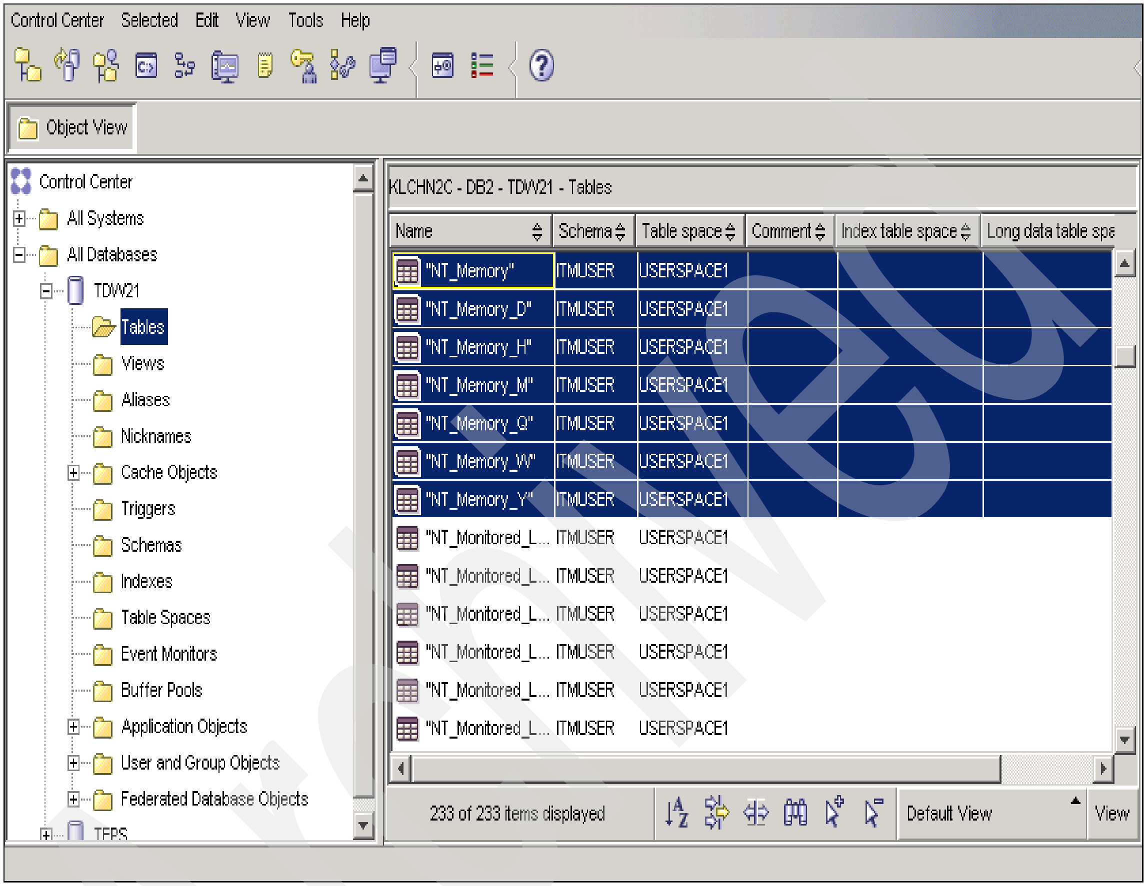
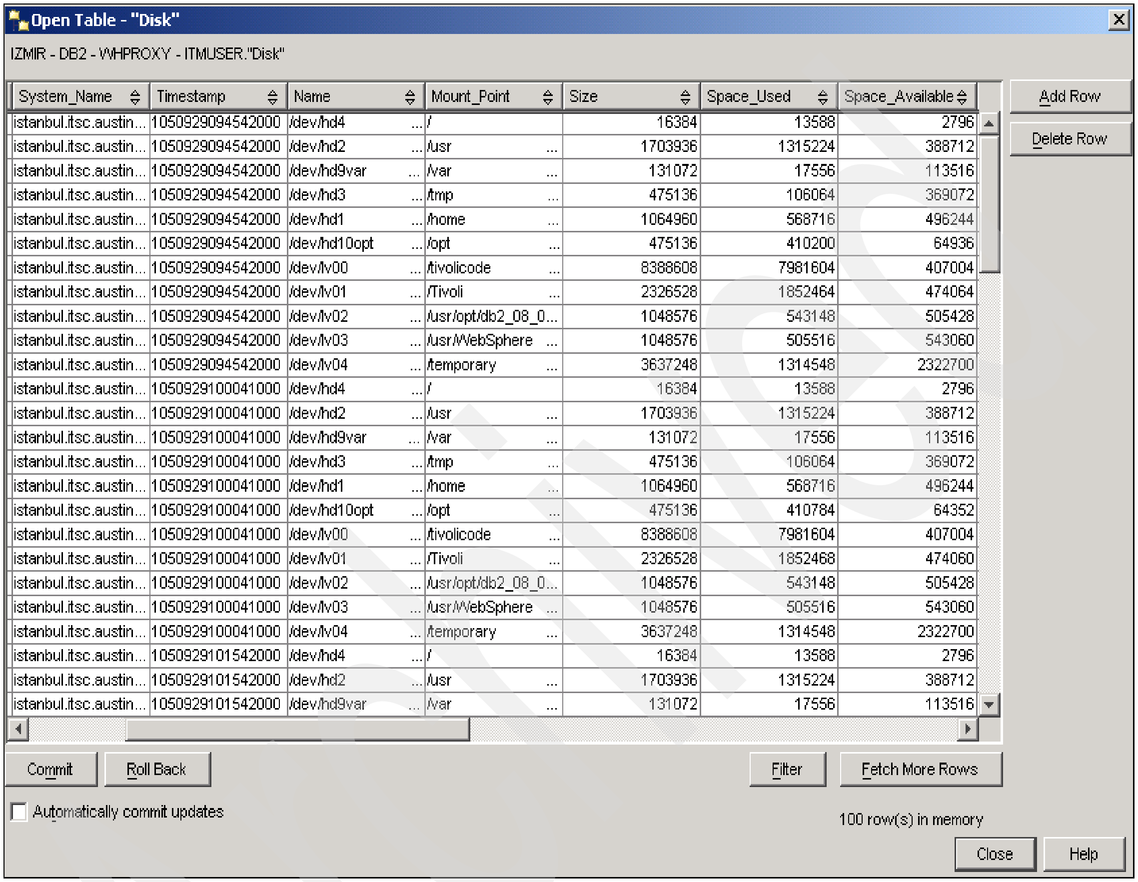


Figure 3-4 Example of NT_Memory Detail and Summarization tables

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 3-5 shows an example of the UNIX Disk attribute group with collected data in the warehouse. Notice that the 1050929094542000 time stamp has 11 file systems for that one collection (cycle).



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 3-5 UNIX "Disk" table (multiple-instance) example

Figure 3-1 on page 114 shows a detailed list of the historical database 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 Tivoli Monitoring Express V6.1 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, 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 to which the attribute group applies. 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 3-5 on page 112 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 Tivoli Monitoring Express V6.1 specific agent monitoring guides. 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 historical database 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 time stamp 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 time stamp 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 historical database 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 explained here.

The origin node can be of the 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 IBM WebSphere MQ, for example).
- *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 historical database is to query the Tivoli Enterprise Portal Server database. On the Tivoli Enterprise Portal Server database machine or from an ODBC connection, you can run the SQL statements shown in Figure 3-1 to get a list of all the installed detailed tables from DB2.

Example 3-1 Connect to teps

```
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, additional tables that include summarized data will be created in the historical database. 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 creates the summarized tables and performs 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. This process sleeps and wakes up every five minutes to check whether the summarization and pruning run has been scheduled to start. (The default schedule is once per day at 2:00 a.m.) If the summarization and pruning is scheduled to run within this five minute interval, it then starts the Summarization and Pruning agent scheduled run against the historical 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 removes 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

The names of the summarization tables are 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 3-3 on page 116 provides a Linux CPU tables example.

Table 3-3 Linux CPU tables example

Time span	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 attributes 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.

The behavior characterization types are:

- GAUGE

  These are 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 are 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

  These are 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 are 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, a delta-based calculation computes the difference between each cycle interval. At the end of the summarization period, it totals all deltas, stores the high value, stores the low value, and stores the last value recorded. *IBM Tivoli Monitoring Administrator's Guide*, SC32-9408, has more details about delta-based summarization.

- PROPERTY

  These are attributes that rarely change (for example, total amount of memory or CPU speed). There is 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

  These are attributes that are high-water marks or snapshot based. There is 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

  These are attributes that are low-water marks or snapshot based. There is 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

  These are attributes used to calculate the number of intervals that are sampled to get an average. There is one attribute in the summarized table using the original attribute name prefixed with just SUM.

- PDEL

  These are attributes that are deltas precalculated by the application (change over a period of time). These attributes are aggregated with MIN, MAX, and SUM values.

- STATE

  This is not used at this time. Generally, this is an enumeration list of options referring to the condition of a resource (for example, up or down).

> **Note:** If a column name exceeds the RDBMS name length (for example, DB2 is 30 characters), the historical database creates an internal column name and stores the internal name and original attribute name in a table called WAREHOUSEID.

For more information about attribute definitions, see Section 4.2.5 of the IBM Redbook *Getting Started with IBM Tivoli Monitoring 6.1 on Distributed Environments*, SG24-7143.

3.3 Planning: Logical configuration considerations

In this section, we discuss the physical and logical planning considerations for implementing the historical database. Before reading this section, review Chapter 2, "Product architecture and deployment best practices" on page 13.

The primary considerations when planning the logical configuration and implementation of the historical database are the estimation of the database size and planning for the kind of data you want to keep in the historical database and how long you want to keep it (pruning).

Database sizing

The IBM Tivoli Monitoring Express V6.1 historical 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 historical database 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 Tivoli Monitoring Express V6.1, all attribute groups relate to tables in the historical database. 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.

Table 3-4 shows the default pruning values for the detailed and summarized tables.

Table 3-4 Default pruning values for detailed and summarized tables

Table	Prune time span
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 DB2 Control Center Estimate Size option).
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 on which each attribute group will collect historical data.
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 100 servers and keep seven days of detailed data, we have 5021 MB (approximately 5 GB) allocated for the Windows OS detailed tables in the historical database. Figure 3-6 illustrates this example.

Attribute group	Bytes/per	Ins	Total1	Intervals	Total2	Servers	Size/day	Days	Total size (meg)
NT_Logical Disk	340	5	1700	96	163200	100	16320000	7	108.9477539
NT_Memory	344	1	344	96	33024	100	3302400	7	22.04589844
NT_Physical Disk	196	3	588	96	56448	100	5644800	7	37.68310547
NT_Process	760	50	38000	96	3648000	100	364800000	7	2435.302734
NT_Processor	192	3	576	96	55296	100	5529600	7	36.9140625
NT_Services	1212	30	36360	96	3490560	100	349056000	7	2330.200195
NT_System	792	1	792	96	76032	100	7603200	7	50.75683594
									5021.85059

Figure 3-6 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 is in the IBM Tivoli Monitoring 6.1 agent user guides (online version shipped with the product) or from an RDBMS tool (such as DB2 Control Center Estimate Size option).

- 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 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 multiplied by the total number of estimated instances for that attribute group.

- Intervals

  This is 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 affect the overall size of the historical 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

  This is the total number of servers from which this attribute group will be collected.

- Size/day

  This is the total size per day for a specific attribute group. For NT_Process, it is 3648000 * 100 = 2435 MB.

- Days

  This it the 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 3-6 on page 120, the total estimated size for the database is 5021 MB (approximately 5 GB). This example uses only the seven attribute groups for the 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 requirements.

Summarized table size calculations

To estimate the summarized tables, 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, we used the DB2 Control Center “Estimate Size” option.

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, the summarized table should use the same estimation for the number of instances. In this example, the “NT_Process_D” table 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 Table 3-4 on page 119, the total number of summarized records is 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 on which each attribute group will collect historical data.

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 3-6 on page 120 and also use Table 3-4 on page 119 for the default summarization pruning values for the same 100 servers, we come up with a total of 26468 MB (approximately 26 GB) for all of the summarization tables, as shown in Figure 3-7.

Attribute group	Bytes/per	Instances	Total1	Summary records	Total/grp	Servers	Total (meg)
NT_Logical Disk	827	5	4135	2668	11032180	100	1052.11067
NT_Memory	1423	1	1423	2668	3796564	100	362.068558
NT_Physical Disk	690	3	2070	2668	5522760	100	526.691437
NT_Process	1236	50	61800	2668	164882400	100	15724.411
NT_Processor	646	3	1938	2668	5170584	100	493.105316
NT_Services	1045	30	31350	2668	83641800	100	7976.70364
NT_System	1308	1	1308	2668	3489744	100	332.807922
							26468

Figure 3-7 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 collecting historical data. Each attribute group creates multiple summarization tables in Tivoli Data Warehouse. There is 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 is 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 using 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 122.

- Total/grp

  This is the total number of bytes per system for each attribute group.

- Servers

  This is the 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.

For our simple example of 100 servers and seven Windows OS attribute groups, we reach a total of 5 GB for the detailed tables and 26 GB for the summary tables for a total of 31 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 historical database:

- The amount of detailed data you keep in the historical database
- The amount of hourly summarized data you keep in the historical database
- Using shift data in the historical database
- Collecting data for multiple instance attribute groups

The amount of detailed data you keep in the historical database

The example in Figure 3-6 on page 120 uses seven days of detailed data for 100 Windows servers and the total is 5 GB. If we simply change that value to 30 days of detailed data, the total size for all seven attribute groups becomes 21522 MB or approximately 21 GB (Figure 3-8 on page 125). The number 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 historical database 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 historical database. Therefore, setting something like NT_Memory to five minutes might be acceptable. However, setting NT_Process to less than 15 minutes might 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	100	16320000	**30**	466.9189453
NT_Memory	344	1	344	96	33024	100	3302400	**30**	94.48242188
NT_Physical Disk	196	3	588	96	56448	100	5644800	**30**	161.4990234
NT_Process	760	50	38000	96	3648000	100	364800000	**30**	10437.01172
NT_Processor	192	3	576	96	55296	100	5529600	**30**	158.203125
NT_Services	1212	30	36360	96	3490560	100	349056000	**30**	9986.572266
NT_System	792	1	792	96	76032	100	7603200	**30**	217.5292969
									21522.2168

Figure 3-8 Detailed data kept for 30 days

Hourly summarized data you keep in the historical database

In the examples used in Figure 3-6 on page 120 and Figure 3-7 on page 123, the summarized data space is more than five times as large as the detailed data space required (26 GB versus 5 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 3-9 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	100	484.25484
NT_Memory	1423	1	1423	**1228**	1747444	100	166.64925
NT_Physical Disk	690	3	2070	**1228**	2541960	100	242.4202
NT_Process	1236	50	61800	**1228**	75890400	100	7237.4725
NT_Processor	646	3	1938	**1228**	2379864	100	226.96152
NT_Services	1045	30	31350	**1228**	38497800	100	3671.4363
NT_System	1308	1	1308	**1228**	1606224	100	153.18146
							12182

Figure 3-9 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 3-6 on page 120, the total space required would be three times the original 155 GB example (76 GB). The key consideration for shift data being collected is the need for collecting shift data. Are there SLA/SLOs in your organization that require reports and data analysis to be done on a shift basis? If the answer is yes, it is a great feature. If the answer is no, you can cut your disk space requirements by more than half if you turn this off.

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 the historical database, all processes that are running on a system will be collected to the database. If we use the example in Figure 3-6 on page 120 and Figure 3-7 on page 123 and we turn off NT_Process and NT_services for historical collection, the numbers are reduced from a total of 31 GB to a total of approximately 3 GB. See the totals in Figure 3-10 and Figure 3-11 on page 127.

The calculation is as follows:

> 256 MB detail + 2766 MB summary = 3022 MB total
> (approximately 3 GB)

Attribute group	Bytes/per	Ins	Total1	Intervals	Total2	Servers	Size/day	Days	Total size (meg)
NT_Logical Disk	340	5	1700	96	163200	100	16320000	7	108.9477539
NT_Memory	344	1	344	96	33024	100	3302400	7	22.04589844
NT_Physical Disk	196	3	588	96	56448	100	5644800	7	37.68310547
NT_Process	760	0	**0**	96	0	100	0	7	0
NT_Processor	192	3	576	96	55296	100	5529600	7	36.9140625
NT_Services	1212	0	**0**	96	0	100	0	7	0
NT_System	792	1	792	96	76032	100	7603200	7	50.75683594
									256.347656

Figure 3-10 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	100	1052.1107
NT_Memory	1423	1	1423	2668	3796564	100	362.06856
NT_Physical Disk	690	3	2070	2668	5522760	100	526.69144
NT_Process	1236	**0**	0	2668	0	100	0
NT_Processor	646	3	1938	2668	5170584	100	493.10532
NT_Services	1045	**0**	0	2668	0	100	0
NT_System	1308	1	1308	2668	3489744	100	332.80792
							2766.8

Figure 3-11 Summarized tables with NT_Process and NT_Services turned off

An alternative to completely turning off the NT_Process and NT_Services attribute groups is to lower the number 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.

We base all of the examples listed earlier on the Windows OS attribute groups. However, similar calculations can be performed for UNIX and Linux attribute groups. Figure 3-12 is an example of using some Linux OS attribute groups. Here again, we see that the process group (Linux_Process) takes up almost 95% 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	50	2073600	7	13.84277344
Linux_Disk	476	10	4760	96	456960	50	22848000	7	152.5268555
Linux_Process	**1116**	**100**	**111600**	**96**	**1.1E+07**	50	**535680000**	**7**	**3576.049805**
Linux_VM_Stats	124	1	124	96	11904	50	595200	7	3.973388672
									3746.39282

Figure 3-12 Sample Linux OS monitoring attribute groups

Table 3-5 lists other multiple-instance attribute groups of which to be aware.

Table 3-5 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

Table	Product	Estimated instances
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, 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 any 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.

3.4 Configuring

There are two parts to configuring the historical database in IBM Tivoli Monitoring Express V6.1: configuring the Summarization and Pruning agent default parameters and configuring the specific agent attribute groups from the Tivoli Enterprise Portal History configuration icon.

3.4.1 Configuring the Summarization and Pruning agent

When IBM Tivoli Monitoring Express V6.1 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, the defaults for all agent default attribute groups can be reconfigured. We recommend that the Summarization and Pruning agent not be started and scheduled to run before the first time defaults are configured. Complete a thorough review of 3.3, “Planning: Logical configuration considerations” on page 118 before configuring the Summarization and Pruning agent default settings.

Figure 3-13 shows how to reconfigure the Summarization and Pruning agent settings from the Manage Tivoli Enterprise Monitoring Services console (right-click the agent and select **Reconfigure**).

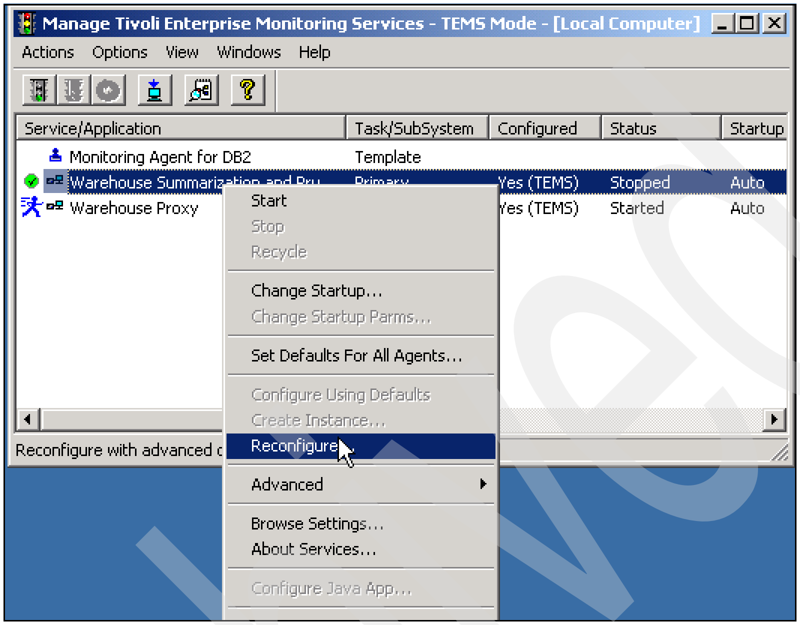


Figure 3-13 Summarization and Pruning agent configuration

Figure 3-14 is an example of the Summarization and Pruning Agent Defaults setting tab. 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.

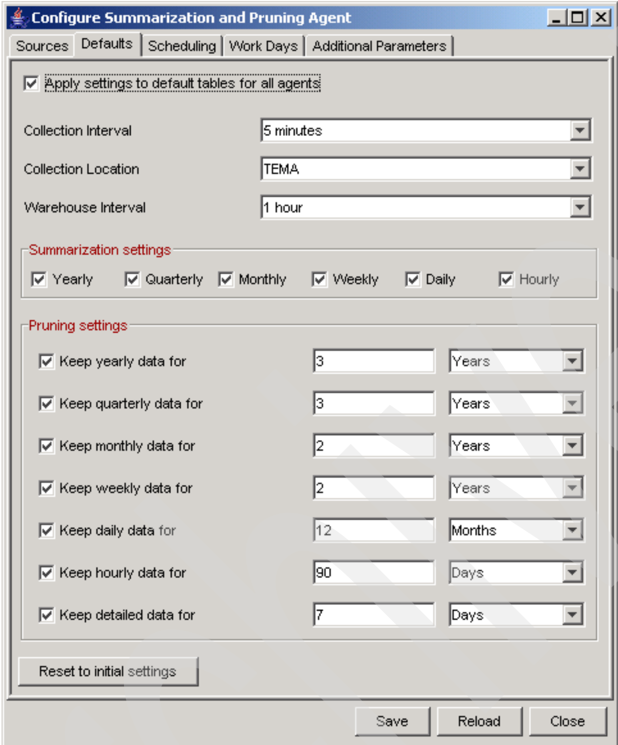


Figure 3-14 Summarization and Pruning Agent Defaults configuration tab

The following list describes the fields in the Defaults tab in Figure 3-14:

- 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 tab. After the summarization and pruning scheduled run has completed, changes to this tab 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 at which the Tivoli Enterprise Monitoring Agent or Tivoli Enterprise Monitoring Server binary data will be uploaded to the Warehouse Proxy agent. The options are one hour or daily. For environments with a lot of agents, we recommend setting 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 historical database. Data older than the prune settings is removed from the historical database.

Figure 3-15 shows an example of the Summarization and Pruning Agent Scheduling tab.

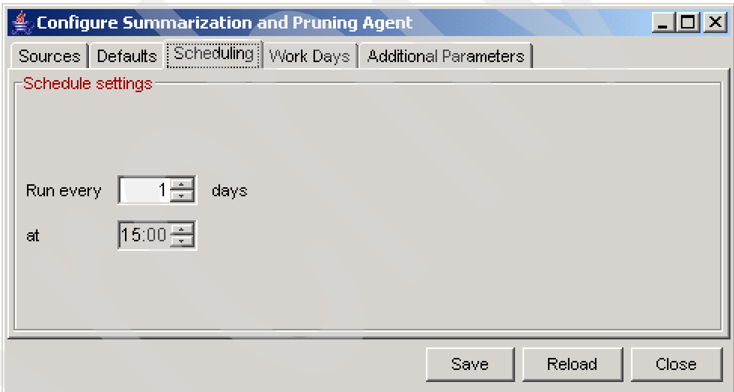


Figure 3-15 Summarization and Pruning Agent Scheduling configuration tab

The fields in Figure 3-15 are:

- Run every

  This option sets the daily cycle time. The default is one day. However, it can be set to run every seven days. We recommend that set summarization and pruning to run every day.

- at

  This value is the time the summarization and pruning run is scheduled every day. The default is 2:00 a.m.

Figure 3-16 shows the Summarization and Pruning Agent Work Days tab.

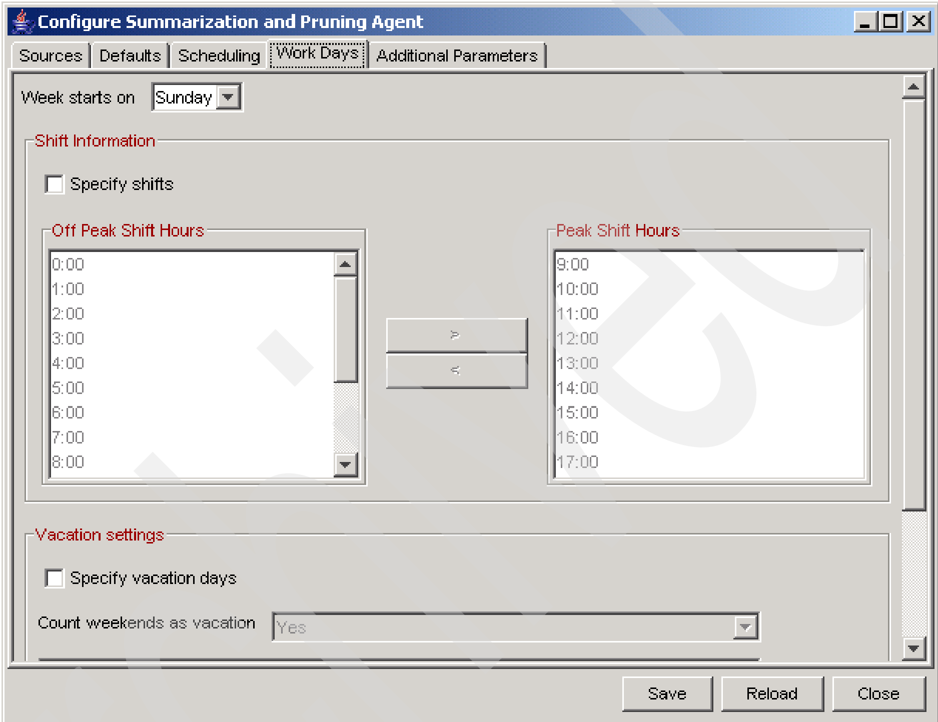


Figure 3-16 Summarization and Pruning Agent Work Days configuration tab

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 enables you to set peak and non-peak shifts. If you select this option, two additional records will be created for each attribute group in the summary tables. Because 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 3-17 shows the Summarization and Pruning Agent Additional Parameters tab.

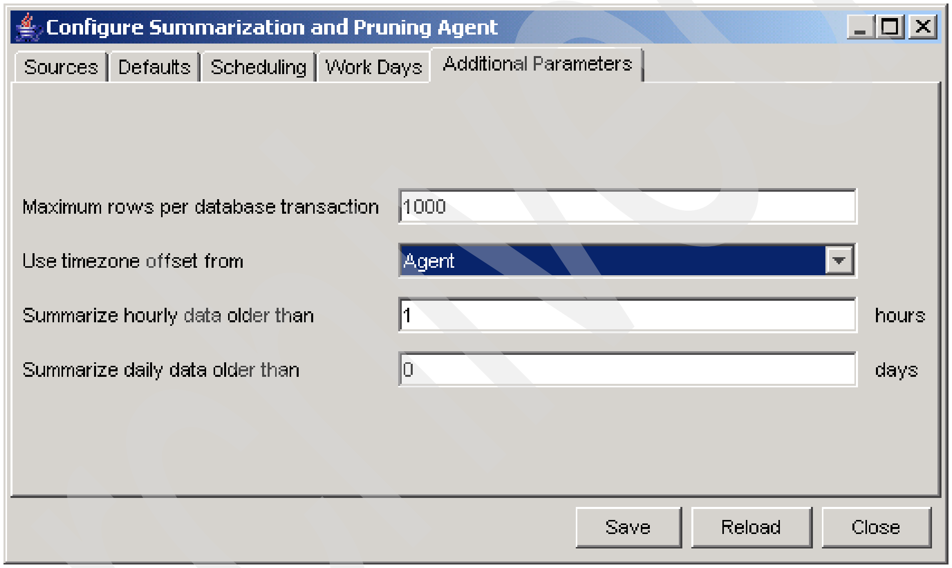


Figure 3-17 Summarization and Pruning Agent Additional Parameters tab

The Additional Parameters tab includes these fields:

- Maximum rows per database transaction

  This specifies the maximum rows that can be deleted in a single transaction.

- Use timezone offset from

  This pull-down list specifies the source for the time zone 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

  This specifies the age of the data you want summarized in Tivoli Data Warehouse. Values are 0 through n. The default is 1 for hourly data and 0 for daily data.

After completing the default Summarization and Pruning agent configurations, start the Summarization and Pruning agent process. The process wakes up every five minutes to check whether it needs to schedule summarization and pruning run. When the summarization and pruning process completes, the defaults are permanent and the ksy.k<pc>.installed files are completed in the logs directory.

3.4.2 History configuration

After the first summarization and pruning process has run, configure the individual agent attribute groups. The agent attribute groups can be configured from the Tivoli Enterprise Portal Server History configuration icon as shown in the steps in Figure 3-18.

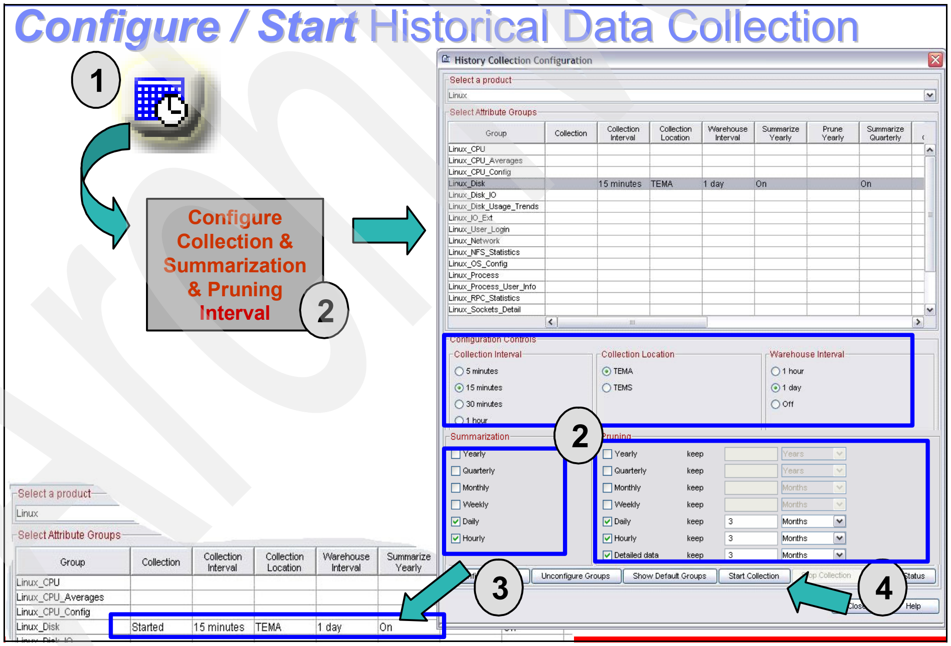


Figure 3-18 History Collection Configuration

Figure 3-18 on page 134 illustrates the following steps:

1. Select the History configuration icon from the Tivoli Enterprise Portal Server GUI.
2. Highlight the specific attribute groups for which you want to collect historical data and add the configuration settings. If you click the **Show Default Groups** button, the panel highlights 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. Click **Configure Groups** for the highlighted groups. If it is the first time you are configuring an attribute group and you selected Show Default Groups, all of the default settings that were defined in the Summarization and Pruning agent configuration are loaded.
4. Highlight the specific groups again and click **Start Collection**.

Figure 3-19 shows an example of configuring the default attribute groups for the Linux OS agent.

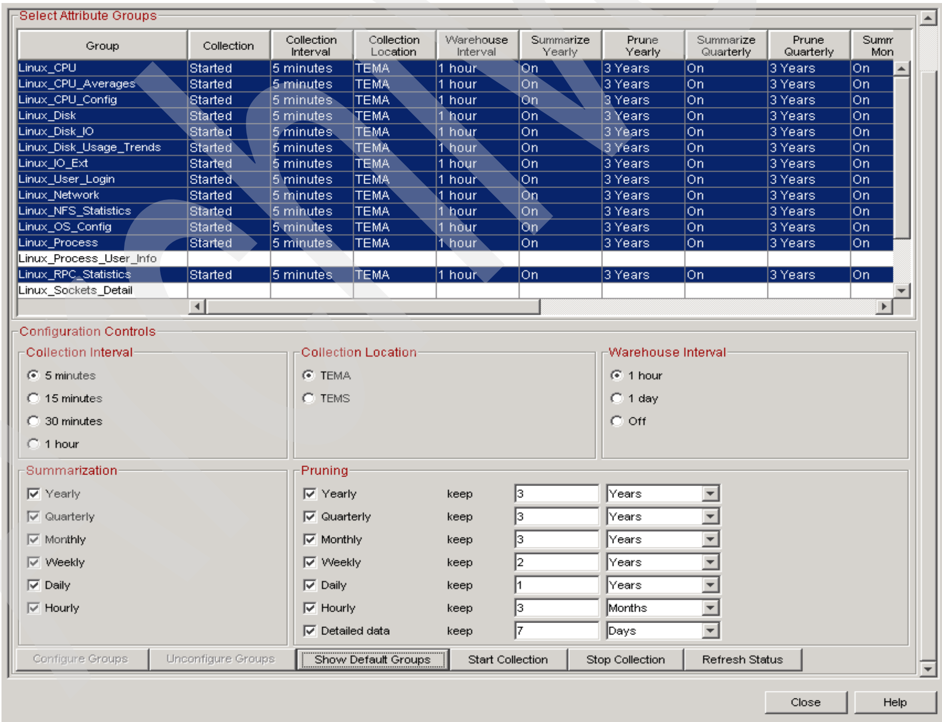


Figure 3-19 History configuration panel

The fields and buttons in Figure 3-19 on page 135 include:

- Collection Interval (radio buttons)

  The collection interval sets the default time to collect data on the Tivoli Enterprise Portal Agent or Tivoli Enterprise Portal Server to the binary files. The default five 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. We recommend that, whenever possible, you select TEMA (at the agent).

- Warehouse Interval (radio buttons)

  This is the interval that the Tivoli Enterprise Portal Agent or Tivoli Enterprise Portal Server binary data will be uploaded to the Warehouse Proxy agent. The options are 1 hour, 1 day, and Off. For environments with a lot of agents, we recommend that you select 1 hour. If you select the Warehouse Interval Off button, no data is collected in the historical database for the selected attribute groups. However, if the attribute group is started with the interval off, the binary data is collected on the agent; however, it never is 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

  This sets how long to keep data in the historical database. Data older than the prune settings will be removed from the historical database.

- 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 unavailable.

- Refresh Status (button)

  Click this to refresh the status (Started or Stopped) of all the agents.

3.5 Reporting: Accessing IBM Tivoli Data Warehouse from Tivoli Enterprise Portal

There are two reporting interfaces you can use to access data in the historical database. The first reporting interface is through the Tivoli Enterprise Portal. Use Tivoli Enterprise Portal 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 Tivoli Data Warehouse database by using a third-party tool. We discuss using Tivoli Enterprise Portal in this section.

IBM Tivoli Monitoring Express V6.1 has seven agents that can collect data into the historical database:

- Microsoft Windows
- UNIX
- Linux
- DB2
- Microsoft SQL
- Oracle
- Sybase

After the agents are configured to collect historical data, all of the reports can be generated from the Tivoli Enterprise Portal Server.

You can view reports from any workspace view by selecting the Time span icon (Figure 3-20). From the Time span window, 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 historical database.

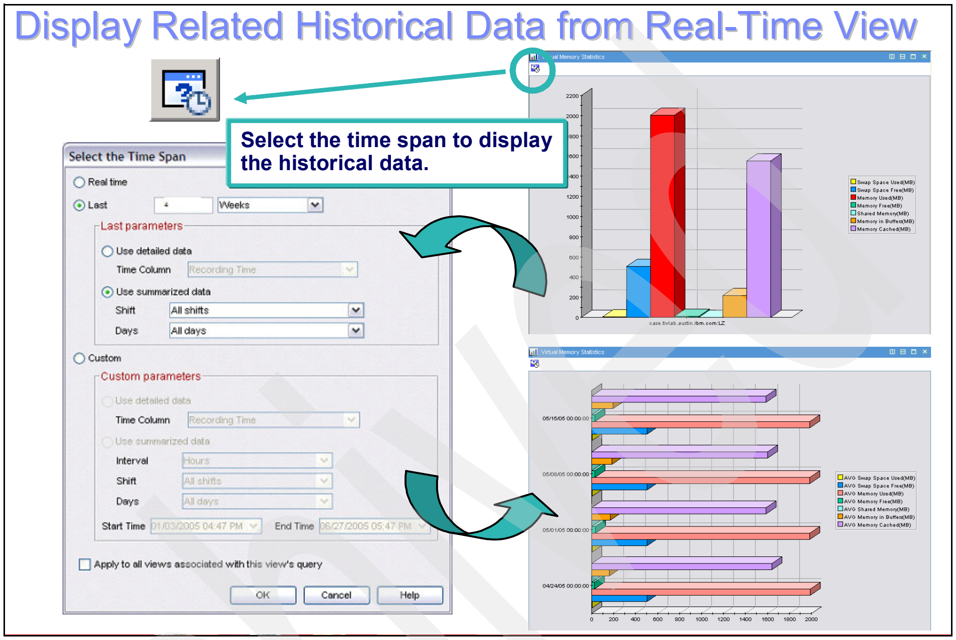


Figure 3-20 Using the Time span icon from a real-time view

Figure 3-21 shows an example of how you can configure the time span view to access the historical database.

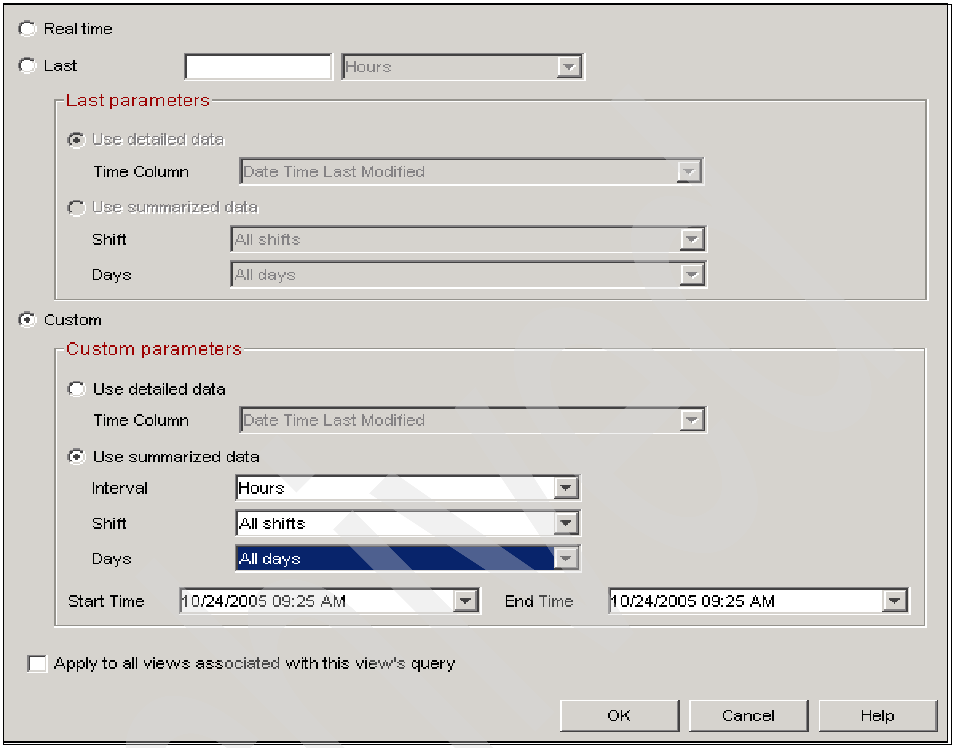


Figure 3-21 Configuring the time span for historical reporting

The fields and buttons in Figure 3-21 include:

- Real time

  This is the default selection that instructs the reporting interface to get data from real-time attributes.

- Last

  If you select this option, the report interface retrieves data from either the binary file tables on the Tivoli Enterprise Monitoring Agent or Tivoli Enterprise Monitoring Server or directly from the data warehouse. You can configure the following additional parameters:

  – Use detailed data

  – Use summarized data

- ► Custom

  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 also ship with historical summarized workspaces. There are three primary types of historical summarized workspaces:

- ► Availability
- ► Capacity
- ► Performance

Figure 3-22 shows an example of the three primary types of historical summarized workspaces views available.

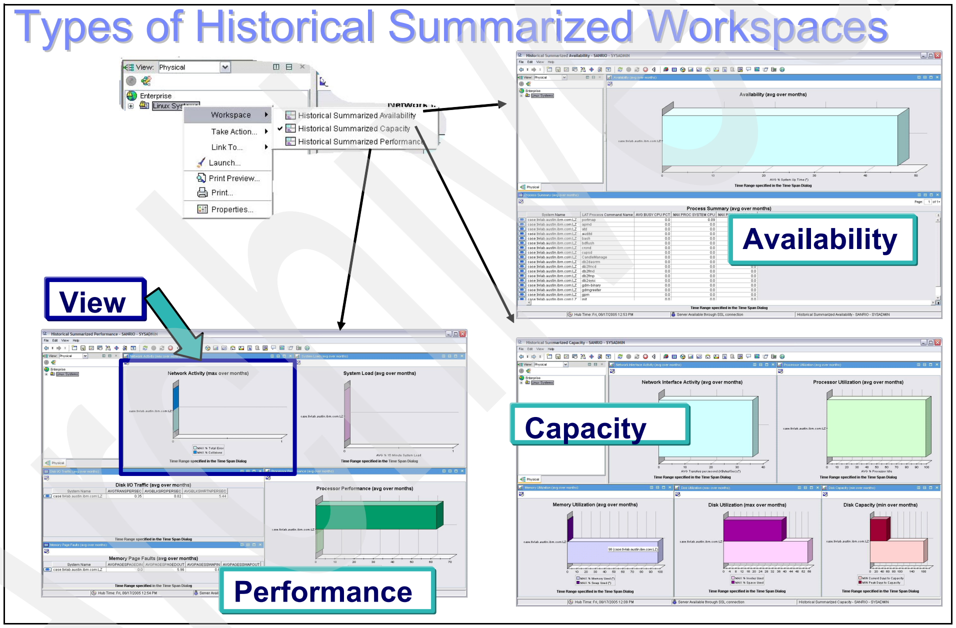


Figure 3-22 Historical summarized workspace views

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 aggregate view (that is, summarized monthly data over the selected time span). From there, you drill down to weekly, daily, and hourly data.

4

Working with IBM Tivoli Monitoring Express V6.1

In this chapter, we demonstrate how to work with the IBM Tivoli Monitoring Express V6.1. First, we describe the Tivoli Enterprise Portal client, and then we show some examples to describe how to use the Tivoli Enterprise Portal client. We include several topics related to user administration. Finally, we discuss IBM Tivoli Data Warehouse.

We discuss the following topics in this chapter:

- Understanding the Tivoli Enterprise Portal client
- Working with Tivoli Enterprise Portal
- Historical data collection
- Solution Installer tool

4.1 Understanding the Tivoli Enterprise Portal client

The IBM Tivoli Enterprise Portal client provides a user interface for IBM Tivoli Monitoring Express V6.1. In this section, we demonstrate how to log on to Tivoli Enterprise Portal and describe the results.

4.1.1 Launching Tivoli Enterprise Portal

As discussed in 2.4.3, “Launching Tivoli Enterprise Portal” on page 29, we can access Tivoli Enterprise Portal as either a desktop application or a Web-based application. The Web-based application is available through Microsoft Internet Explorer, and any workstation that has access to the Tivoli Enterprise Portal Server can access this application.

Refer to the following sections for details about how to launch the Tivoli Enterprise Portal:

- “Launching Tivoli Enterprise Portal from a desktop client” on page 38
- “Launching Tivoli Enterprise Portal from a browser” on page 38

The working area of Tivoli Enterprise Portal is divided into a Navigator and a workspace area.

4.1.2 Tivoli Enterprise Portal components

After logging on to Tivoli Enterprise Portal, the window shown in Figure 4-1 opens.

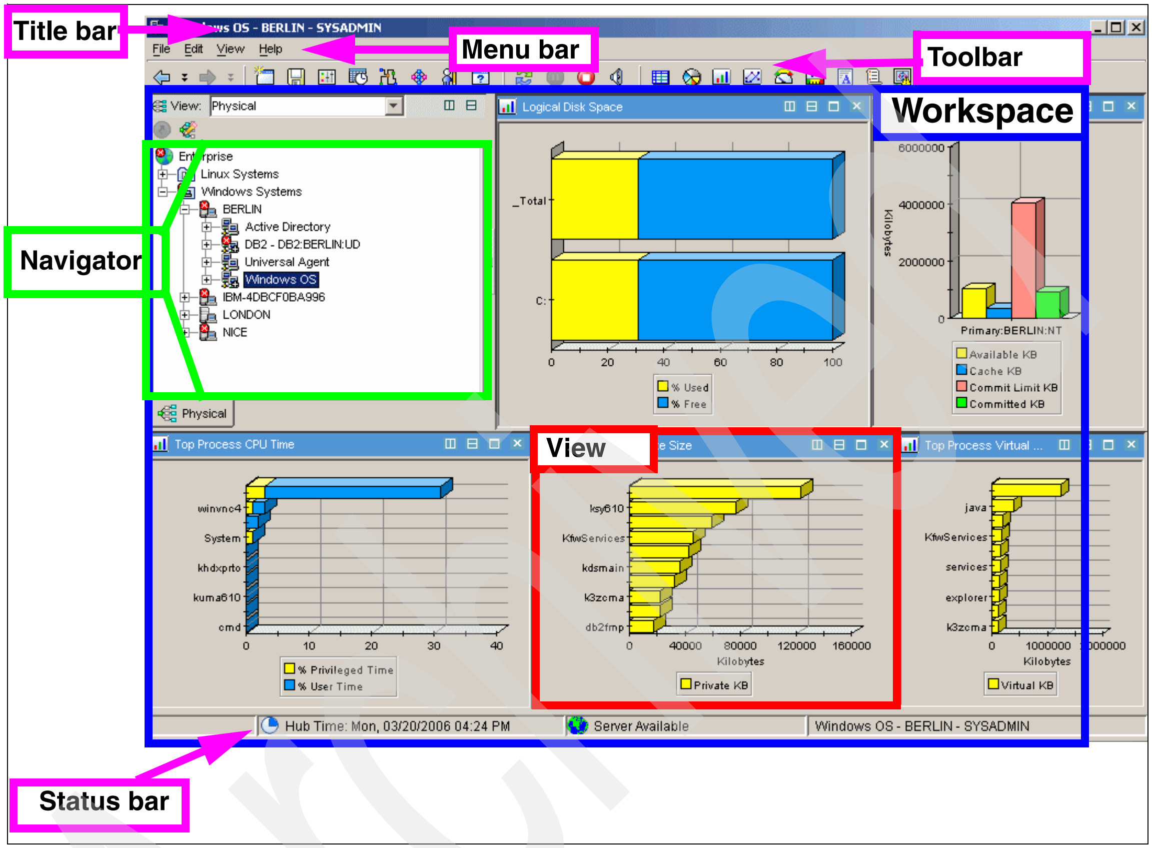


Figure 4-1 Tivoli Enterprise Portal desktop application

The three main components in the Tivoli Enterprise Portal are the Navigator, the workspace, and the views.

Navigator

You can navigate through this tree view (see Figure 4-1) of the monitored environment by clicking items, each of which opens a different workspace. It enables you to structure your enterprise information in a way that is meaningful to your users and to the purpose of the monitoring solution.

The Navigator has two view choices: the physical view and the logical view.

Physical view

This shows the network hierarchy from a system point of view. It is organized by operating platform, system name, monitoring agent, and attribute groups.

Logical view

This enables you to organize your view according your logical hierarchy. For example, you can have a Navigator view for your departments.

Workspace

The monitoring data is displayed in a workspace. The workspace is the working area of the Tivoli Enterprise Portal window. Its panes show different types of views. Every time you select a Navigator item, you change the workspace appearance.

Views

A view is a pane in the workspace that contains data from a monitoring agent such as a chart or a table. There are non-data views such as the browser view and the terminal view.

4.2 Working with Tivoli Enterprise Portal

This section walks you through some examples of working with IBM Tivoli Monitoring Express V6.1 using the Tivoli Enterprise Portal desktop application. The IBM Tivoli Monitoring Express V6.1 product comes with a set of predefined workspaces. This solution provides custom workspaces that are tailored for the use of small and medium businesses. Custom workspaces are provided for:

- Database server (IBM DB2 and Microsoft SQL Server)
- Microsoft Active Directory Server
- IBM HTTP Server and IBM WebSphere Application Server
- Microsoft Windows and Linux operating systems

4.2.1 Creating a new workspace and adding custom views

First, we create a new workspace and add the custom views.

Navigating through workspaces

Navigating means to select or expand the items under the Navigator. When you select or expand an item in the Navigator, its default workspace opens. A Navigator item can have multiple workspaces and it can have links to other workspaces.

Expanding and collapsing the tree

Figure 4-2 shows the first view of the Navigator when you start Tivoli Enterprise Portal.

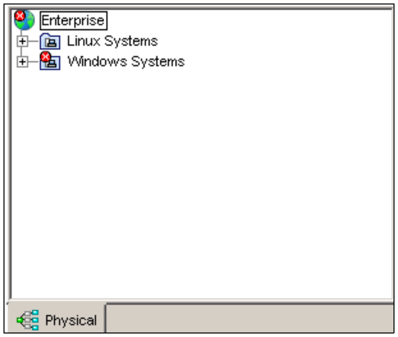


Figure 4-2 The Navigator view

You can access different levels in the Navigator hierarchy by expanding or collapsing the Navigator tree.

The different levels in the physical Navigator are:

- Enterprise
- Operating platform, for example, Windows Systems
- System
- Agent, for example, Windows OS
- (Subagent)
- Attribute group

Expand each level of the Navigator until you reach the lowest level, as shown in Figure 4-3.

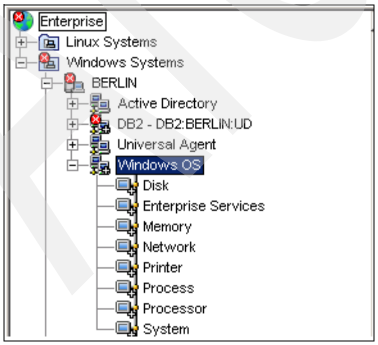


Figure 4-3 The lowest level of the Navigator

Navigating through the workspaces

When you select an item under the Navigator tree, a new workspace opens. The views change each time you select a Navigator item. Select Process to open a workspace with views related to process attributes, as shown in Figure 4-4.

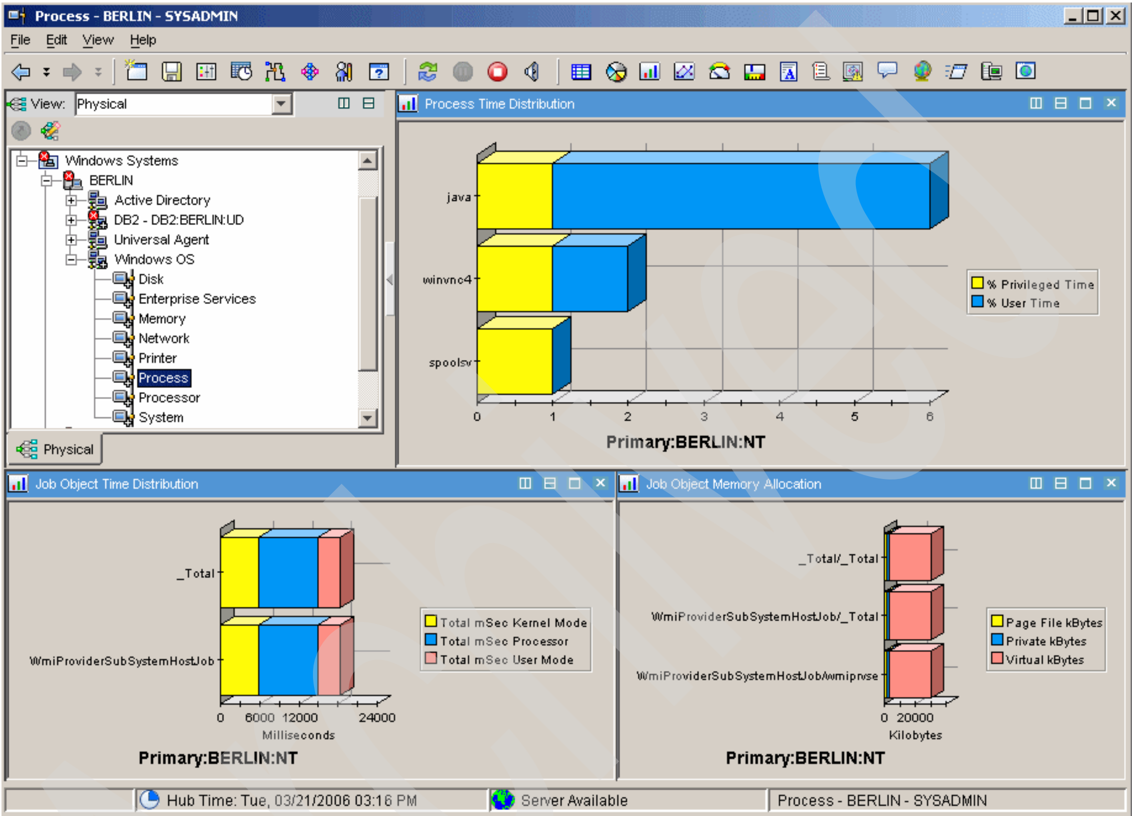


Figure 4-4 Selecting the process attributes

Saving the workspace

Whenever you make a change in the workspace, for example, changing from a bar chart to a table, the system warns you about changing the workspace and asks whether to save the changes.

Perform the following tasks:

1. To create and save a new workspace with the changes, click **Yes** in the Save Workspace message window (Figure 4-5).

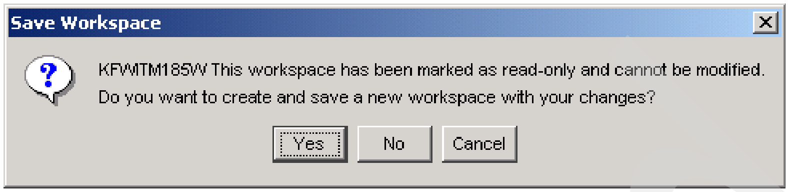


Figure 4-5 Save Workspace message

2. Type the name of this workspace as `NewWorkspace`, type a description, such as `Saving Workspace example`, and click **OK** (Figure 4-6).

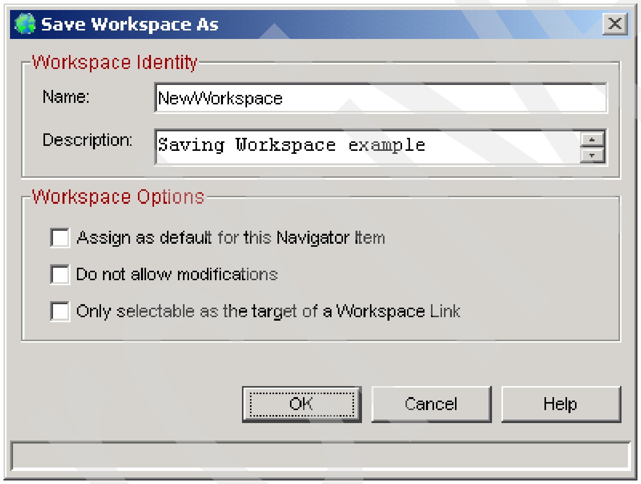


Figure 4-6 Saving the Workspace

You can also save the workspace manually by selecting **File** → **Save Workspace**.

> **Note:** The title bar now shows the name of the saved workspace as NewWorkspace.

Selecting a workspace

The default workspace for the Enterprise Navigator view is called *Enterprise Status Workspace*. To select another workspace from the same Navigator view, perform the following steps:

1. Launch the Tivoli Enterprise Portal desktop client.
2. In the Navigator, right-click **Enterprise** and select **Workspace** → **NewWorkspace**, as shown in Figure 4-7.

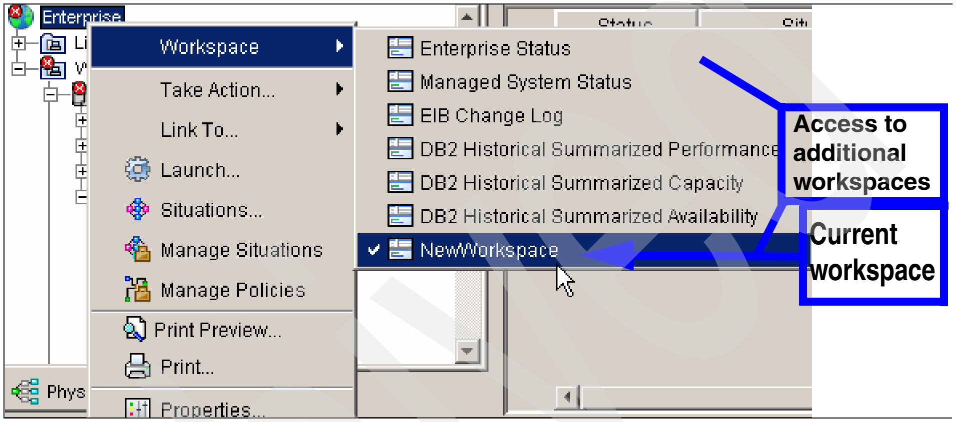


Figure 4-7 Selecting the Workspace

> **Note:** When you expand the Navigator view, you can see that there are other workspaces available.

Working with the views

You can add several types of views in a workspace. In this section, we show you how to add different types of views.

View types

The workspace has the following views:

The table view and the chart views display data that the monitoring agents have gathered from the systems where they are running. They can also show data from any Open Database Connectivity (ODBC)-compliant database for which you write a custom query.

The Notepad view opens a simple text editor for writing text that you can save with the workspace.

The Message Log view shows the status of all situations distributed to the managed systems in your enterprise.

The Situation Event Console view shows the status of all situations associated with items on this branch of the Navigator view and has tools for instant filtering and event handling.

The Universal Message Console view shows the situation and the messages received as the result of universal message generation.

The Graphic view places the Navigator items as icons on a map or a picture of your choice.

The Take Action view enables you to send a command to a managed system.

The Terminal view starts a 3270, 5250, or Telnet session, and enables you to write scripts for working with IBM z/OS applications.

The Browser view opens the integrated browser for accessing Web pages.

You can add as many views to a workspace as you can easily see within the confines of the window.

Adding a non-data view

To add a non-data view, perform the following steps:

1. Open the workspace where you want the view.
2. In the Navigator, expand **Windows Systems**.
3. Select the node of your choice. In our example, we select **BERLIN**.
4. In the toolbar, select the **Situation Event Console** view.

   Note that when you select a view, the mouse pointer changes to a pointing finger.

5. Click the view at the right side of the top pane. This view becomes a Situation Event Console view, as shown in Figure 4-8.

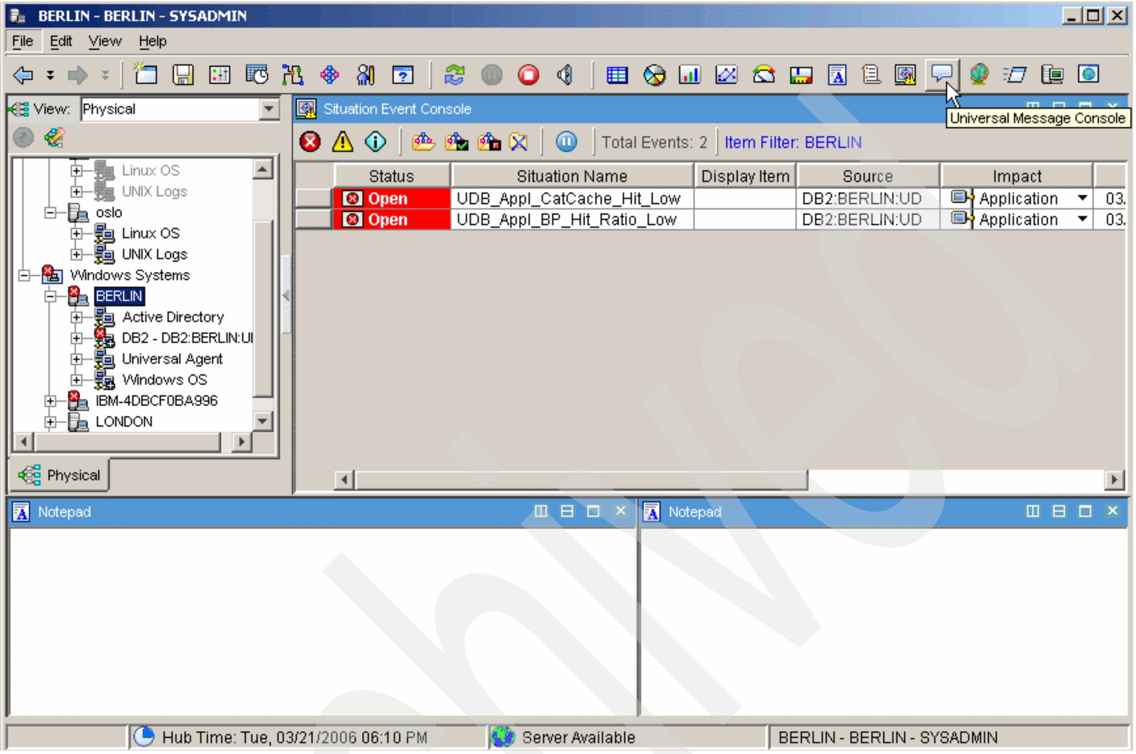


Figure 4-8 Adding a view

6. Click **File** → **Save Workspace** to save the change.

4.2.2 Working with queries

The Chart and Table views show the attribute values from the Tivoli Monitoring Agents or the ODBC data source. The IBM Tivoli Monitoring Express V6.1 products come with queries that are used to populate the table and chart views in workspaces. When you add a table or a chart view over a non-data view (as Message Log or Notepad view) you have to define the query.

Working with data view

This section shows how to add a view that queries for the Monitoring Agent data.

Adding the table view

To add the table view, perform the following steps:

1. In the Navigator, expand **Windows Systems**.
2. Select the node of your choice. In our example, we select **BERLIN**.
3. In the toolbar, click the **Table** view.
4. Click the view at the left side of the bottom plane. This opens the Select option window (Figure 4-9). Click **Yes**.

Figure 4-9 Assigning a query

5. In the Properties - BERLIN window (Figure 4-10), select **Click here to assign a query**.

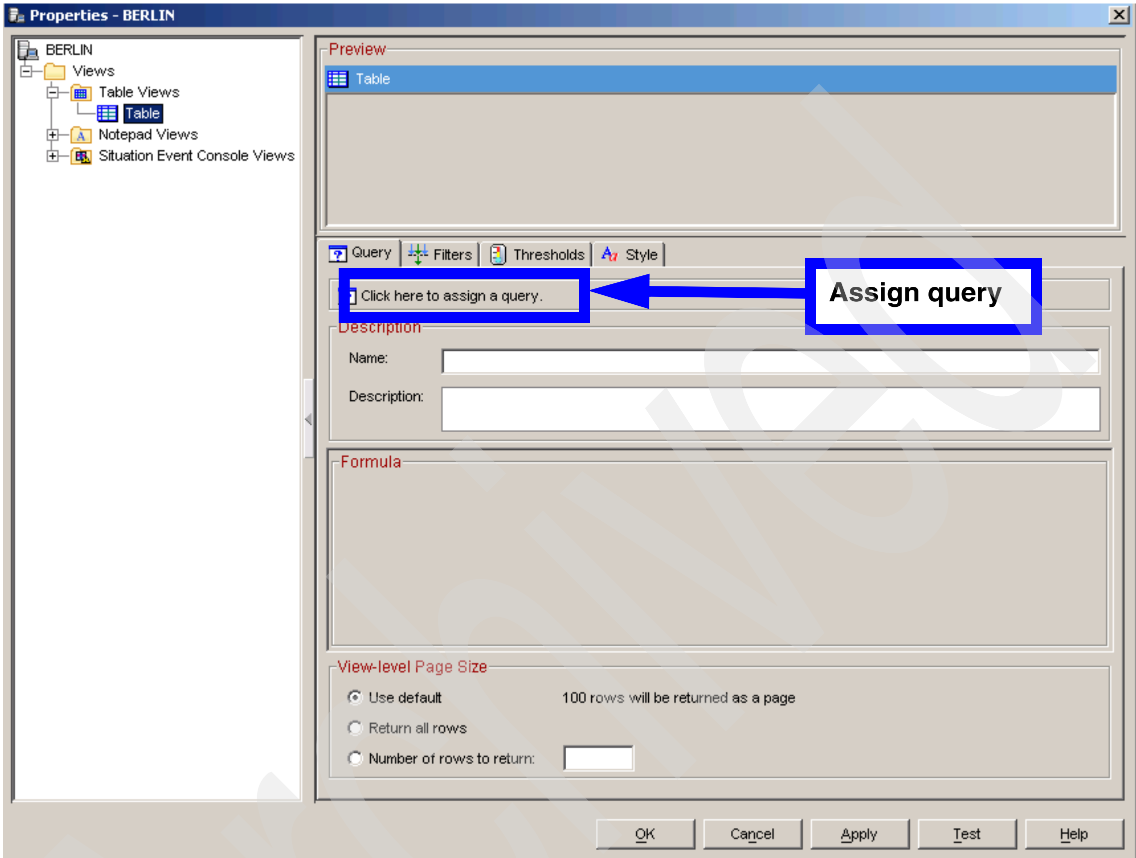


Figure 4-10 Assigning a query

6. The Query Editor opens, as shown in Figure 4-11. Select the **Create Query** icon to open the Create Query window.

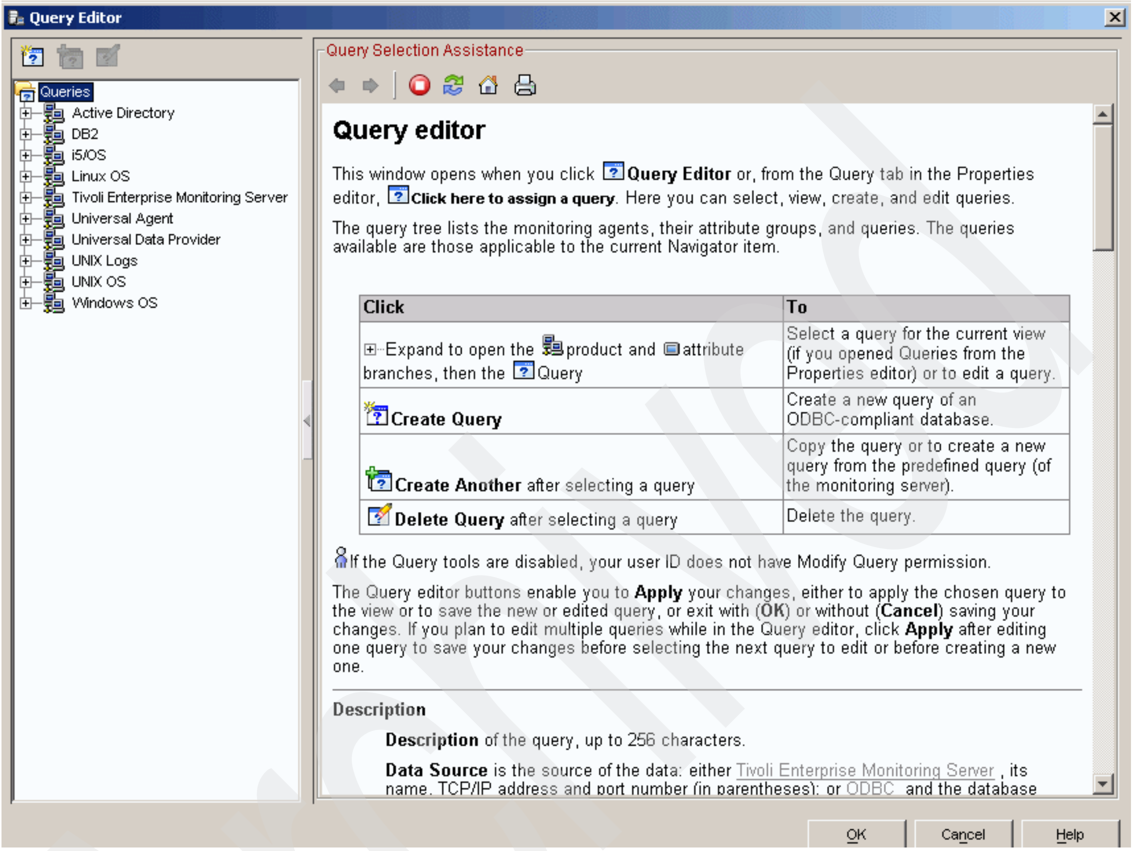


Figure 4-11 Query Editor

7. In the Create Query window (see Figure 4-12), use the following values:

   a. Name the query `Service_status_example`.
   b. In the Description box, type `service status`.

   > **Note:** Changing queries affects every view where this query is used. Be careful when changing queries because it can change the views of other users.

   c. In the Category field, select **Windows OS**.
   d. For Data Sources, select **TEMS**.

   Click **OK**.

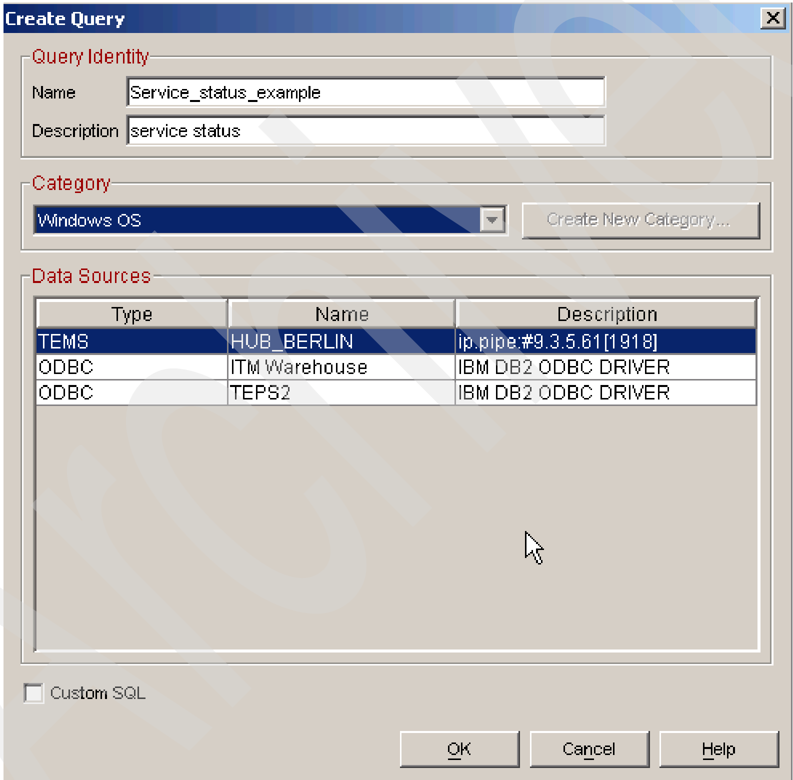


Figure 4-12 Create Query

8. In the Select attribute window, select the attribute group **NT Services**. Press the Ctrl key to select the multiple attribute items **Current State, Display Name, Server Name**, and **Service Name**, as shown in Figure 4-13. Click **OK** to finish selecting the attributes.

Note: Monitoring agents are made up of attributes that represent the properties of systems or networks, such as the amount of CPU usage or the message ID. Attributes are organized into attribute groups. The attributes in a group can be displayed in a table view or chart view or used to specify a condition for testing in a situation. When you open the view or start the situation, data samples are taken from the selected attributes. IBM Tivoli Monitoring Express comes with a set of common attribute groups that can be applied to any managed system.

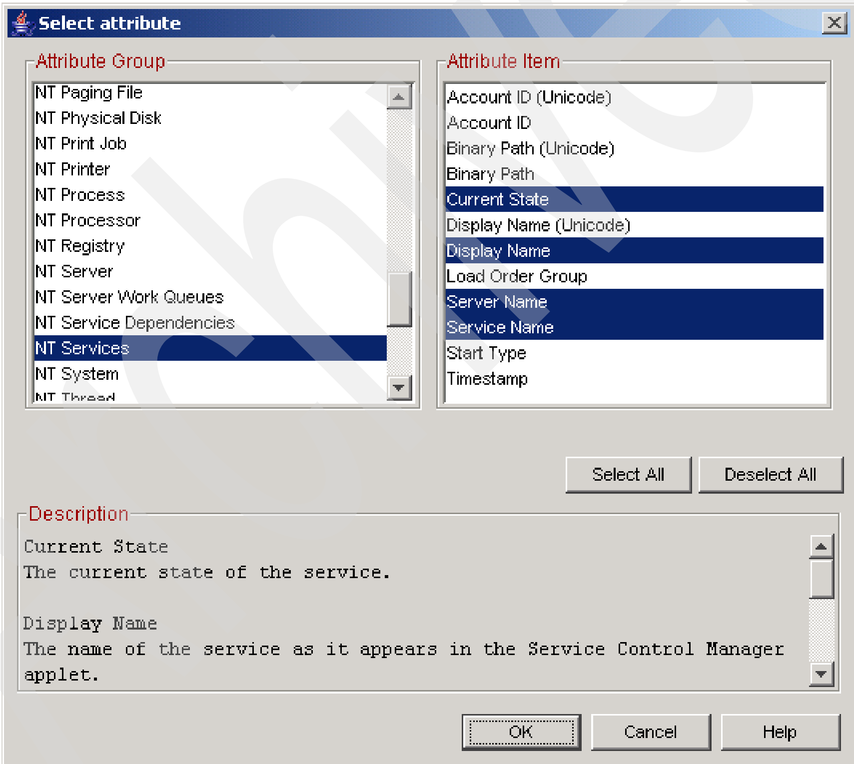


Figure 4-13 Selecting the attributes

9. The new query now appears in the Query Editor (Figure 4-14). You must configure it. Click **OK**.

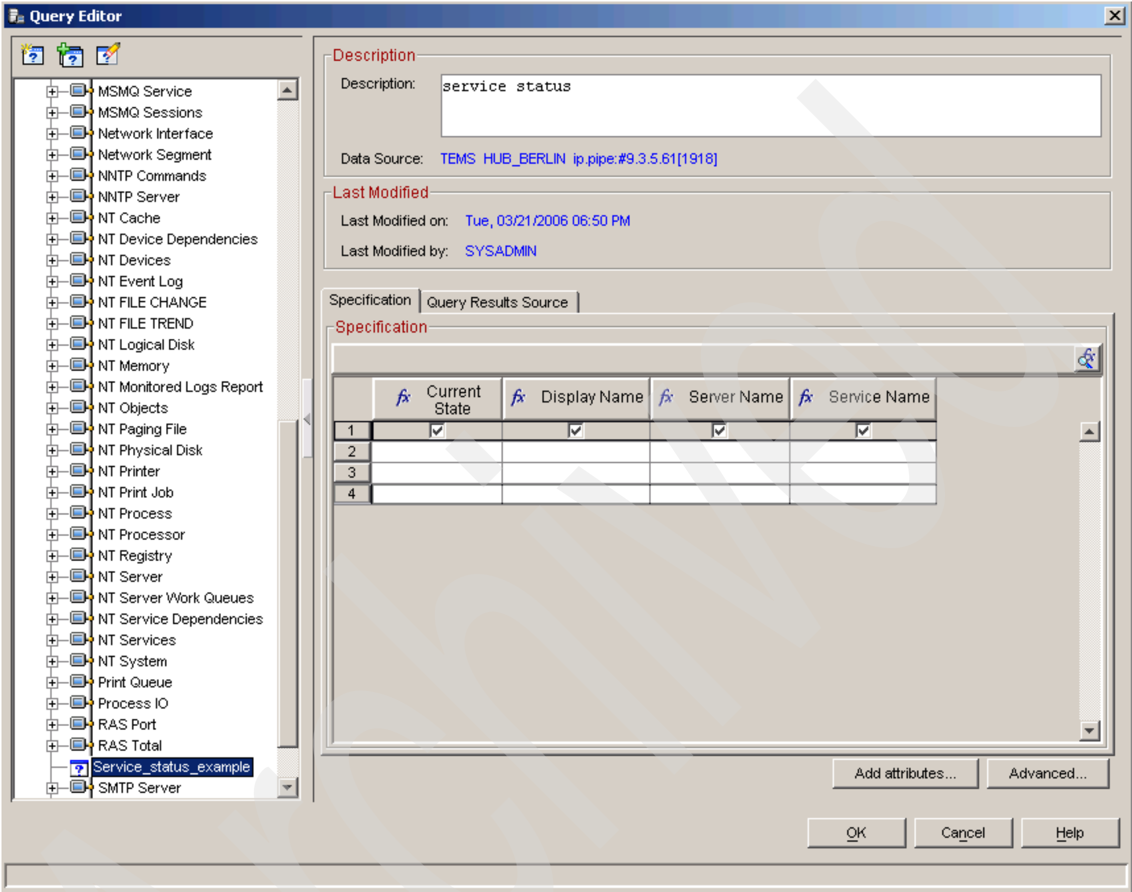


Figure 4-14 New query window

10. Select **Click here to assign a query**. In the Specification field, click the **Server Name** column. Type `$NODE$` and leave the operator as `==` (equal sign), as shown Figure 4-15. Click **Advanced**.

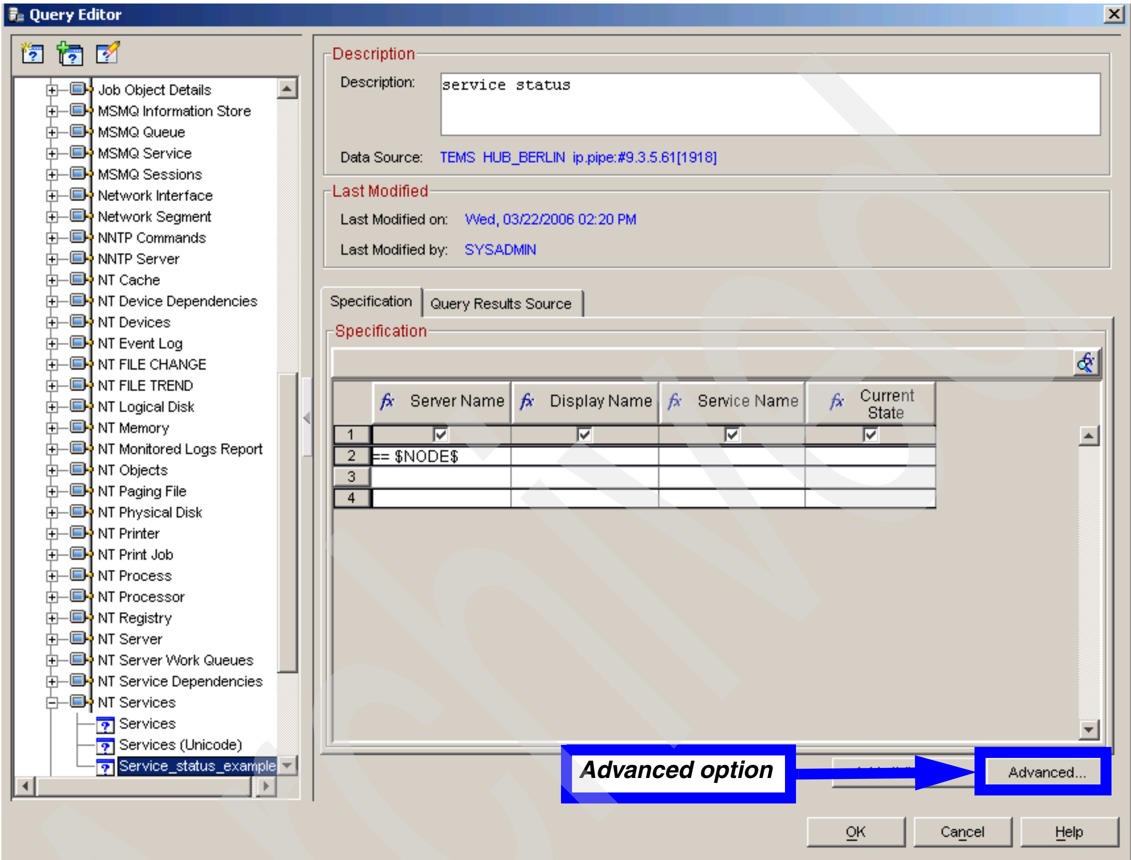


Figure 4-15 Query Editor Specification

11. In the Advanced Options window, select **Display_Name** and **Ascending** (Figure 4-16). Click **OK**.

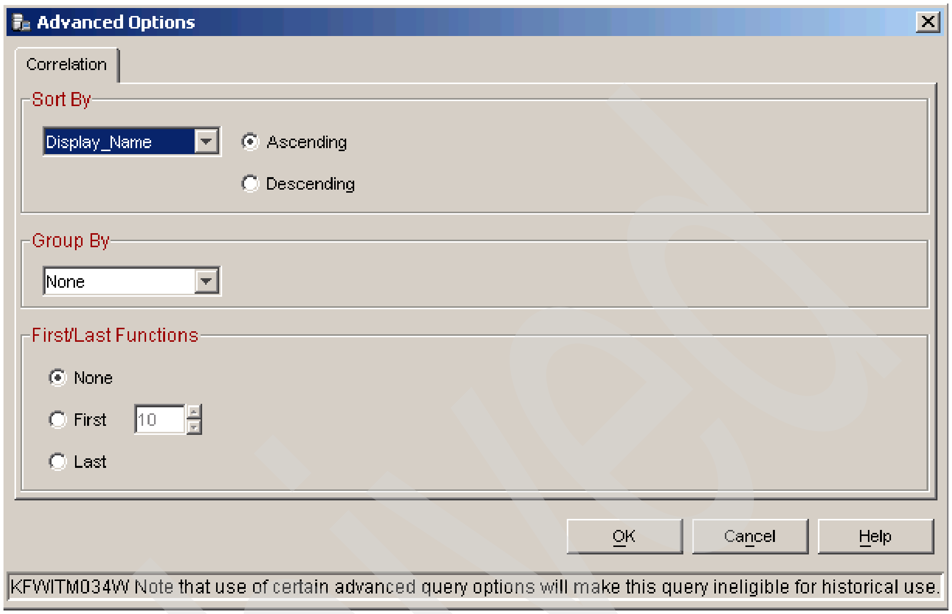


Figure 4-16 Advanced Options

12. Click the **Filters** tab (Figure 4-17). We can filter this to track the status of certain services. Perform these steps:

   a. Click the **Current State** column, change the operator to `==` (equal sign), and type `Stopped`.

   b. Under Display Name, select **==** (equal sign) and type `Telnet`.

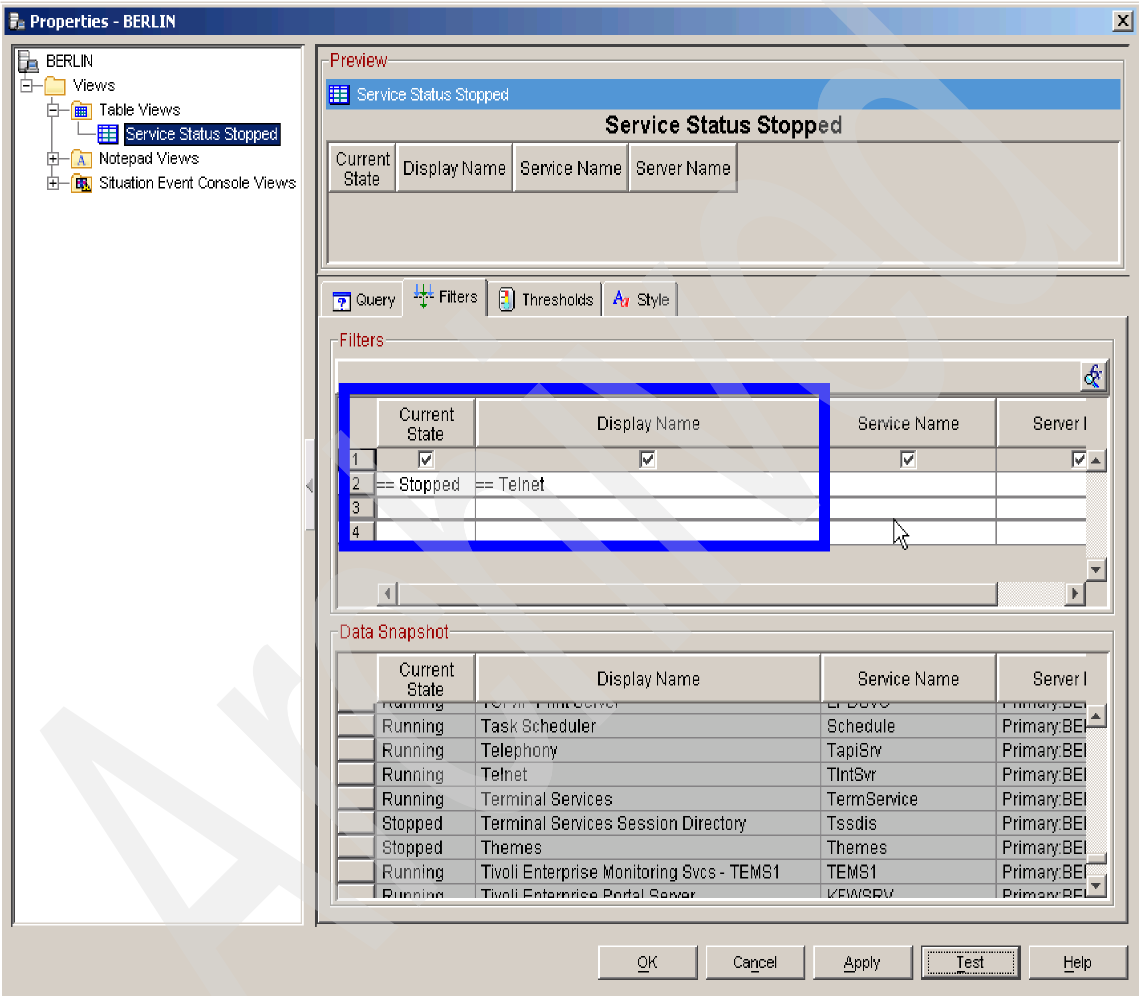


Figure 4-17 Filtering Service Name

13. To change the table name, click the **Style** tab (Figure 4-18). Use the following values:

   a. In the Options field, select the **Show** check box.

   b. In the Title field, type `Service Status Stopped`.

   c. Click **OK**.

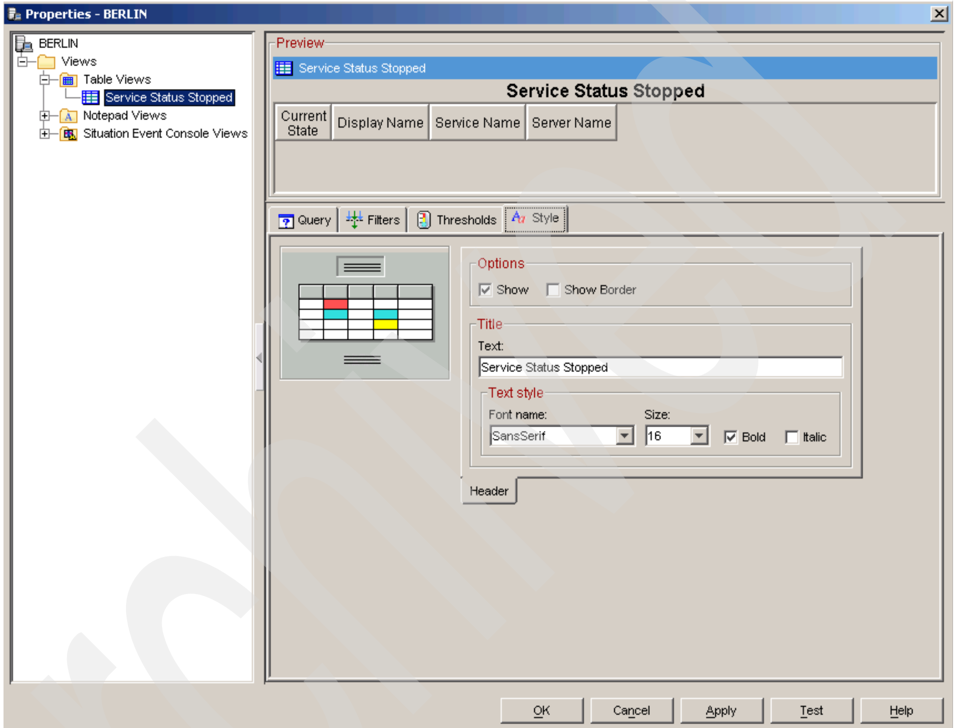


Figure 4-18 Style tab

14.Note that a green icon appears in the Navigator view (Figure 4-19). This means that the updates are pending in the Navigator tree.

Now the Services Status Stopped table view is added to the workspace. Select **File** → **Save Workspace** to save this last configuration.

Because the Telnet service is running, you cannot see any rows in the table view, as shown Figure 4-19.

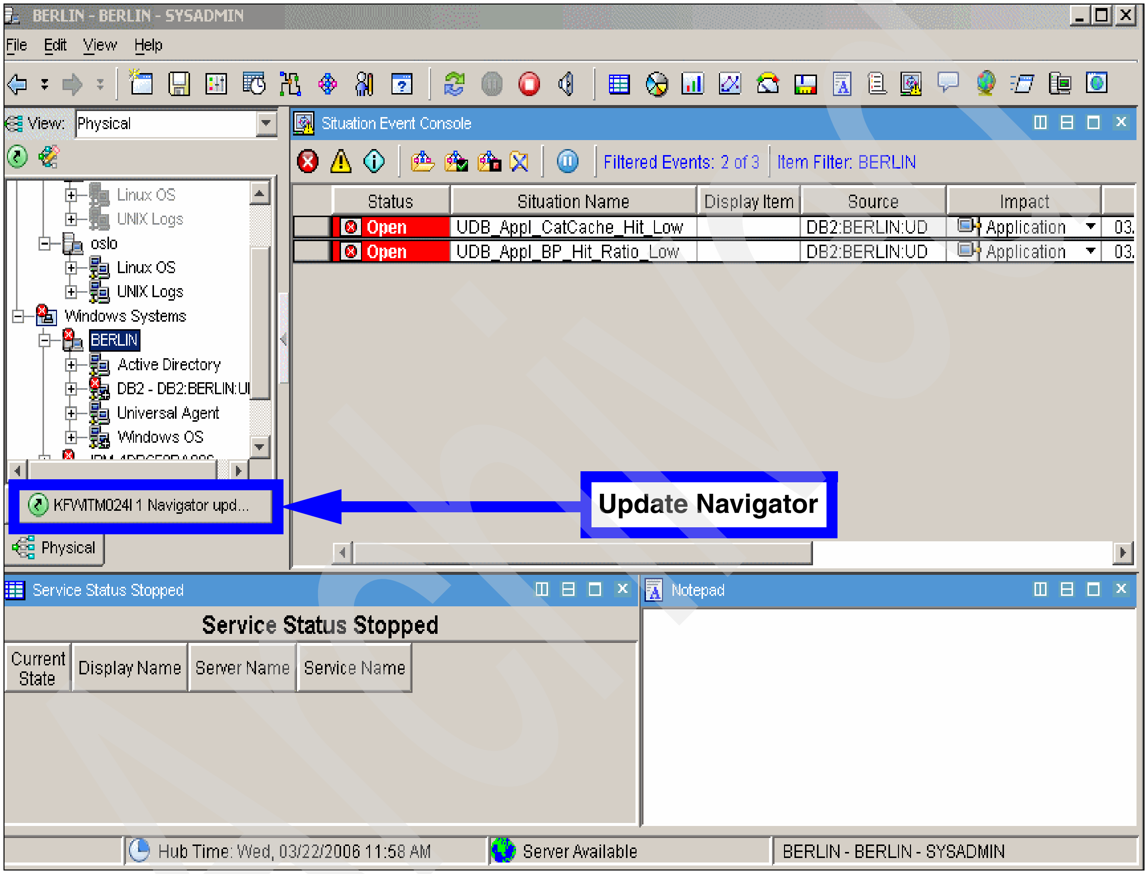


Figure 4-19 New view: Service Status Stopped

You can reproduce the service stopped event to see the view behavior. Perform the following tasks:

1. In Windows, click **Start** → **Run** and type **`cmd`** in the open window. Click **OK**.

2. At the command prompt, type **net stop telnet**. The following message appears:

```
C:\Documents and Settings\Administrator>net stop telnet
The Telnet service is stopping.
The Telnet service was stopped successfully.
```

3. Open the Tivoli Enterprise Portal client (Figure 4-20), which shows the service listed as stopped.

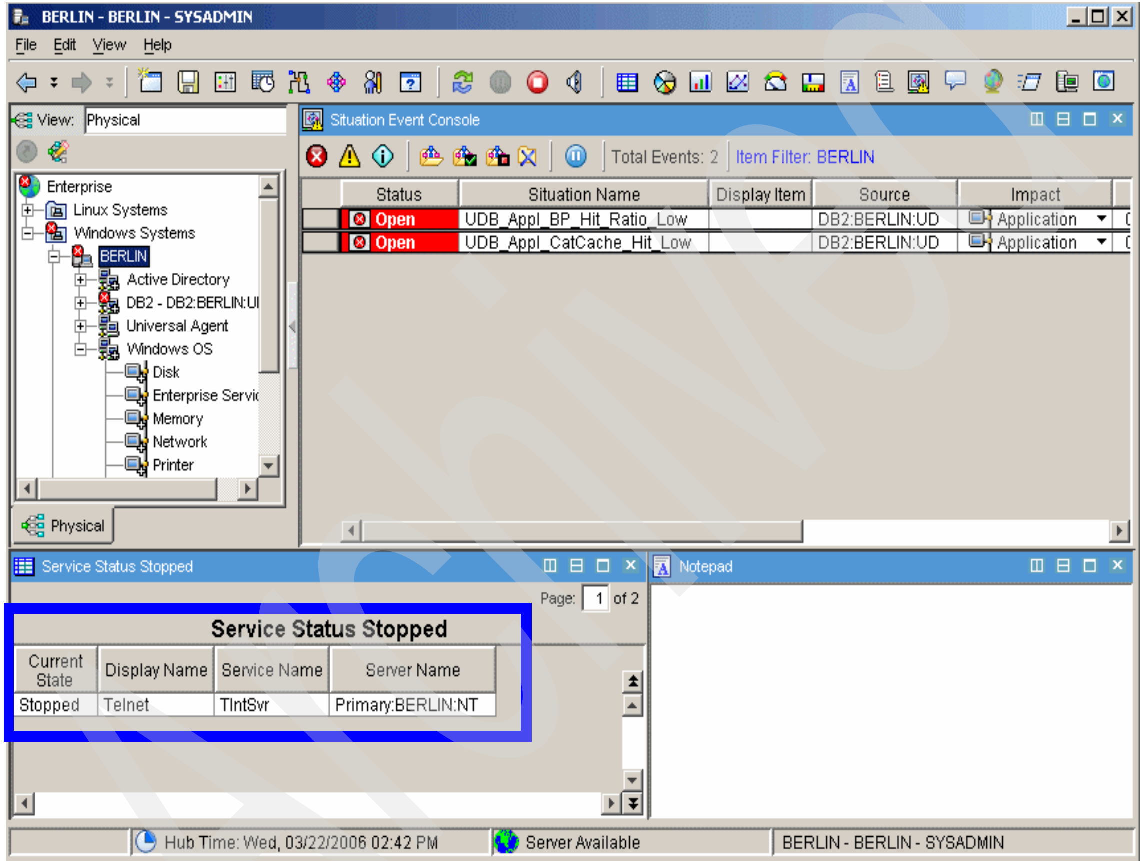


Figure 4-20 Service Telnet Stopped

Adding the chart view

Now that you added the table view, you can add a chart view in the workspace. Perform the following steps:

1. In Tivoli Enterprise Portal, select the **Pie Chart** view and click the right side at the bottom of the pane.

2. When prompted to “Assign query now,” click **Yes**.
3. In the Properties window, select **Click here to assign a query**.
4. In the Query Editor Navigator, expand **Windows OS** → **NT Logical Disk** → **Logical Disk** and click **OK**.
5. Select the **Filters** tab to filter what you want to show in your pie chart, as shown in Figure 4-21.

> **Note:** Unlike queries, using filters does not affect other views.

   a. Select **% Used** and **% Free**.
   b. Under Disk Name, select **==** and type `C:`.

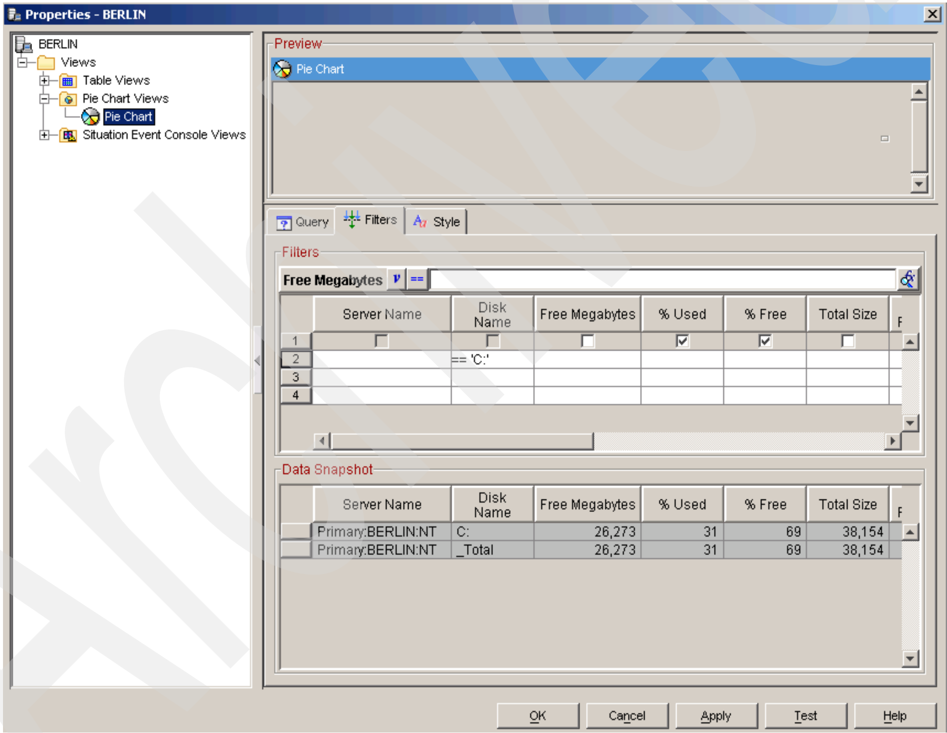


Figure 4-21 Filtering the Pie Chart

6. Select the **Style** tab to name the view. In the Options field, select the **Show** check box and in the text box type `Disk Space`. Click **OK**. This view (Figure 4-22) is added to the workspace.

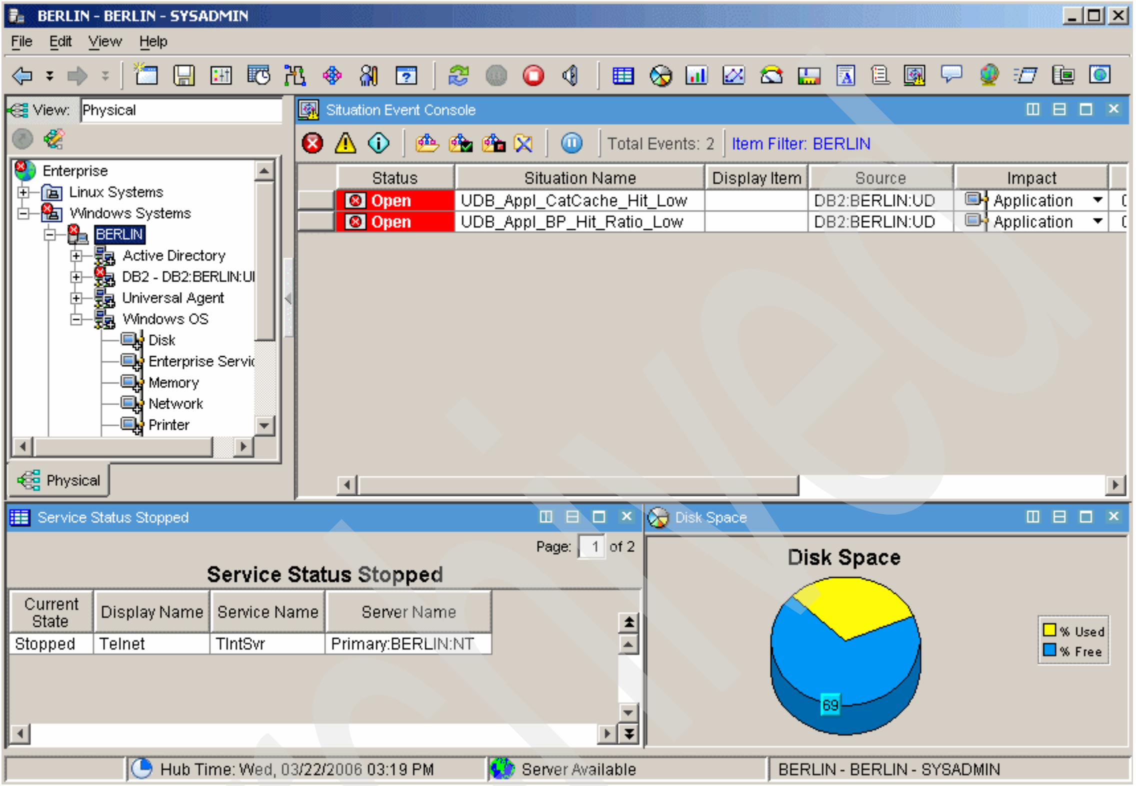


Figure 4-22 Disk Space pie chart view

4.2.3 Working with the Monitoring Agent for Active Directory

You can add a table view with some information about the Monitoring Agent for Active Directory, but before performing this, you have to create another view.

The view toolbar has the following tools for creating the view:

- Split view horizontally
- Split view vertically
- Maximize the view
- Close the view

To divide and add this new view, perform the following steps:

1. To split the Service Status Stopped view horizontally, click .

> **Note:** You have two views with the same query; therefore you should clean the previous query before adding a table view. Add a Notepad view to clean it.

2. Click the **Notepad** view and click the **Service Status Stopped** view. A Notepad view is added, as shown in Figure 4-23.

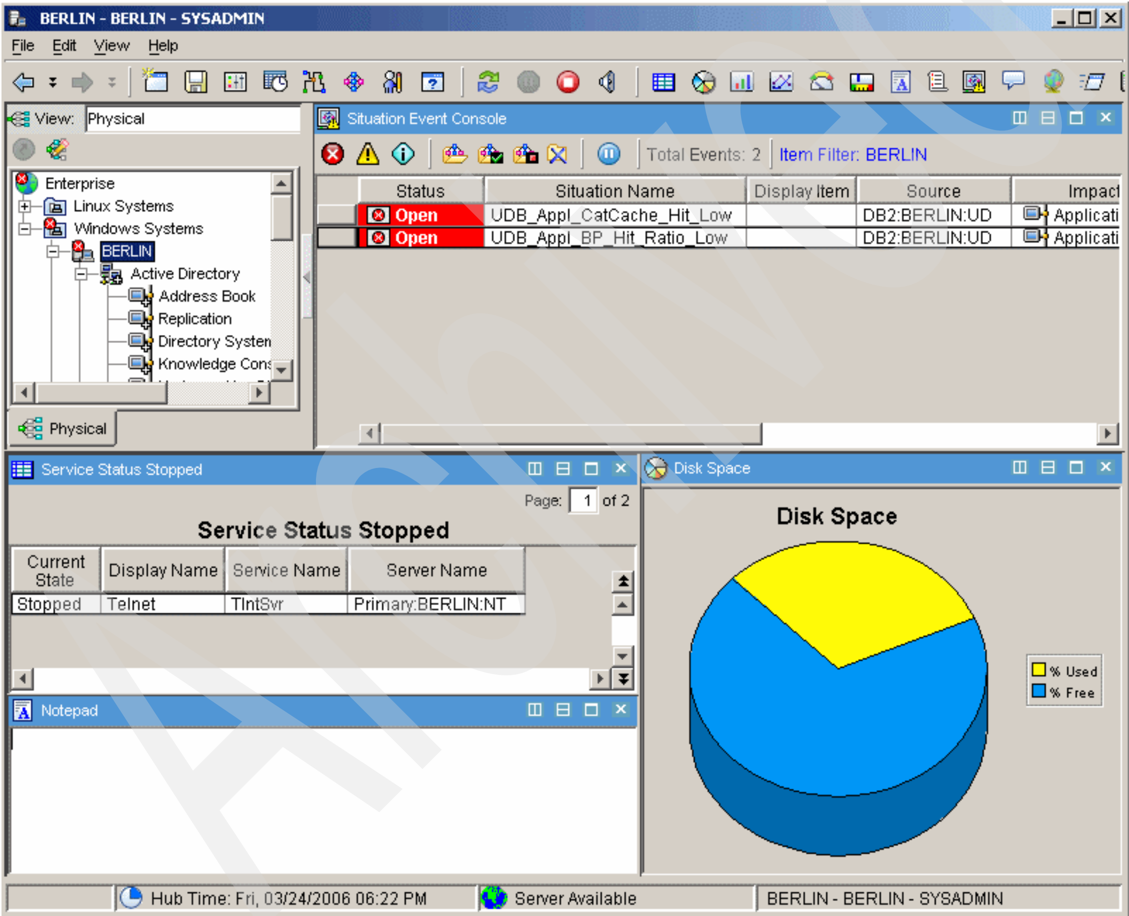


Figure 4-23 Notepad view

3. Select the **Table** view and click the **Notepad** view.
4. For the “Assign query now” question, click **Yes**.

5. In the Properties window, select **Click here to assign a query**.
6. In the Query Editor, expand **Active Directory** → **Domain Controller Availability** and click **OK**.
7. Click the **Filters** tab, and select only **Server Name**, **DCA FSMO Role**, **DCA PDC Master**, and **Timestamp**, as shown in Figure 4-24.

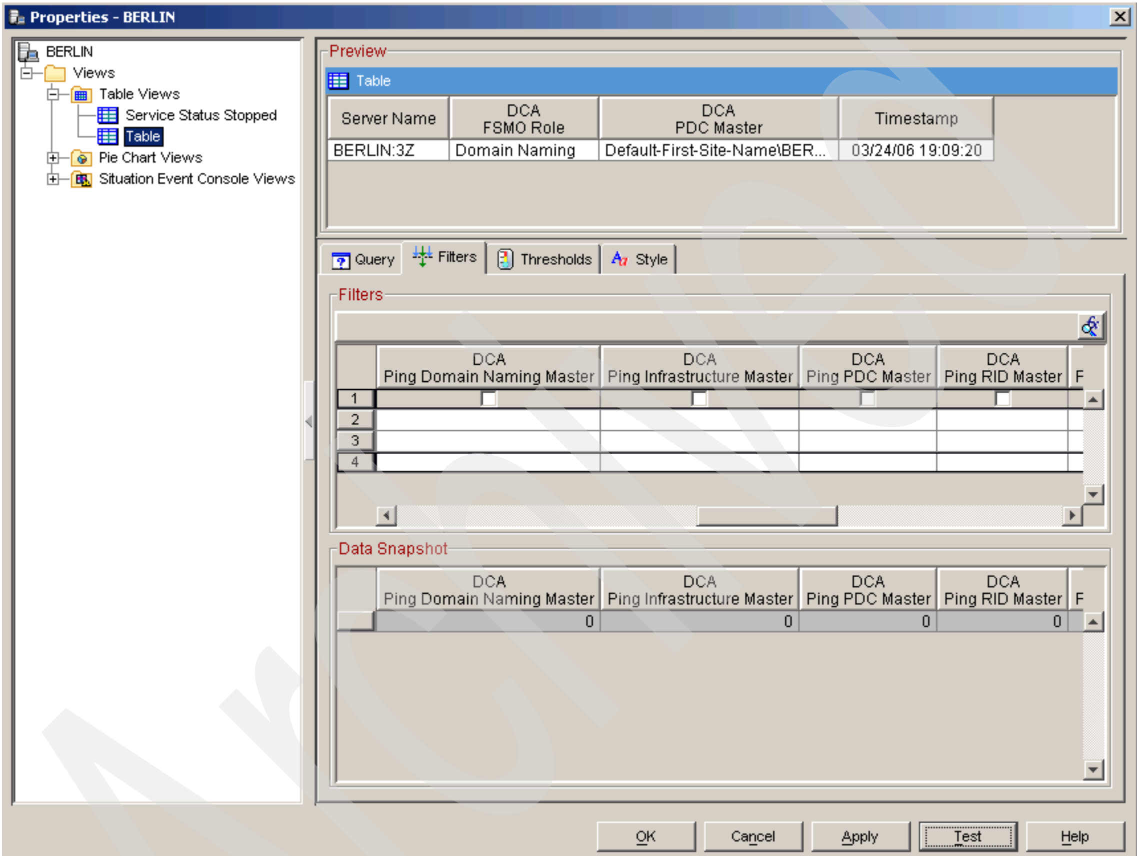


Figure 4-24 Selecting the Filters

8. Click the **Style** tab, select the **Show** check box, and in the Title field, type `Domain Controller Info`. Click **OK**.

   The new view (see Figure 4-25) is added to the workspace.

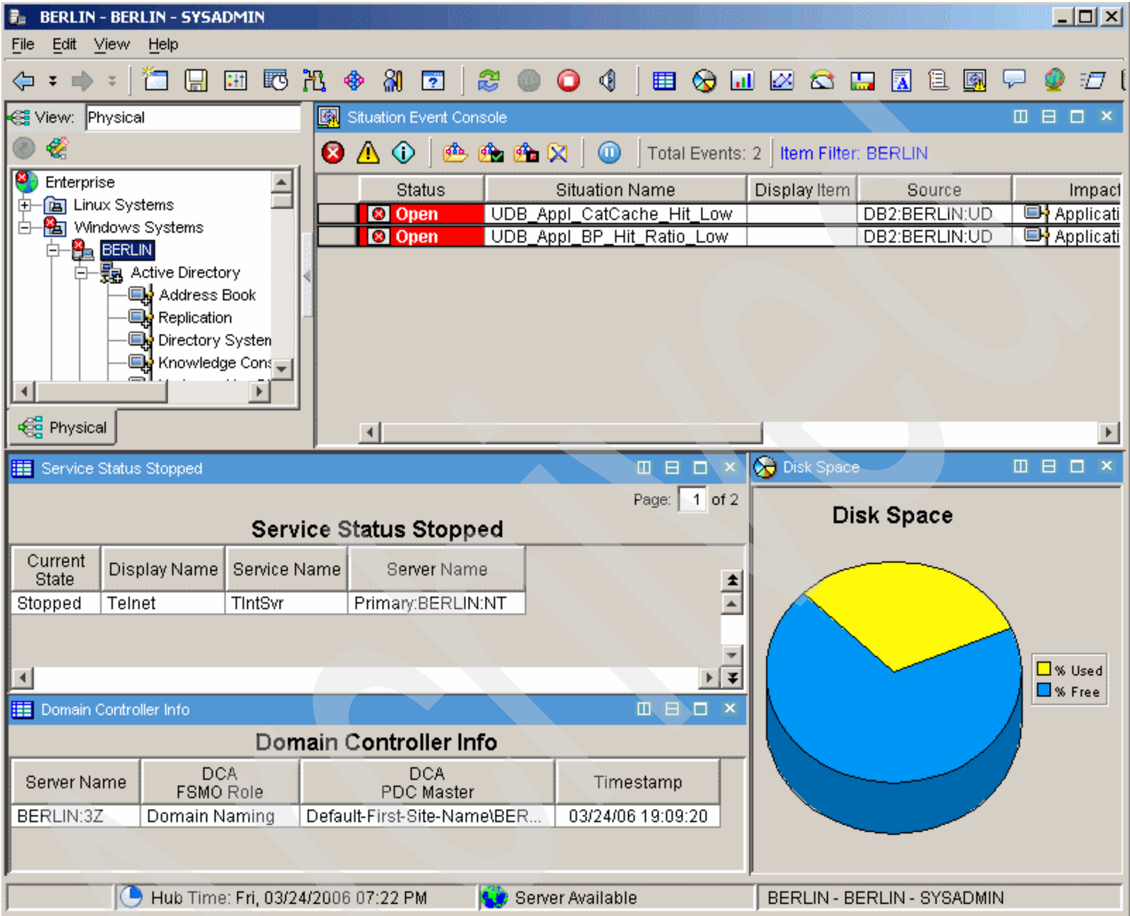


Figure 4-25 Domain Controller Info view

Working with thresholds

In the table view, you can add thresholds to highlight cells whose values meet the threshold set. You can also have thresholds for circular gauge charts and linear gauge charts.

In this example, we split the Domain Controller Info table vertically to add another table view and work with a threshold. Perform the following steps:

1. To split the Domain Controller Info view horizontally, click .

2. Click the **Notepad** view, and click **Domain Controller Info**.
3. Select the **Table** view and click the **Notepad** view.
4. For the “Assign query now” question, click **Yes**.
5. In the Properties window, select **Click here to assign a query**.
6. In the Query Editor, expand **Windows OS** → **NT Process** → **Process Overview** and click **OK**.
7. Click the **Filters** tab, and select only **Process Name**, **%User Time**, and **Timestamp**.
8. Click the **Thresholds** tab and set the Thresholds values for **% User Time** (Figure 4-26).

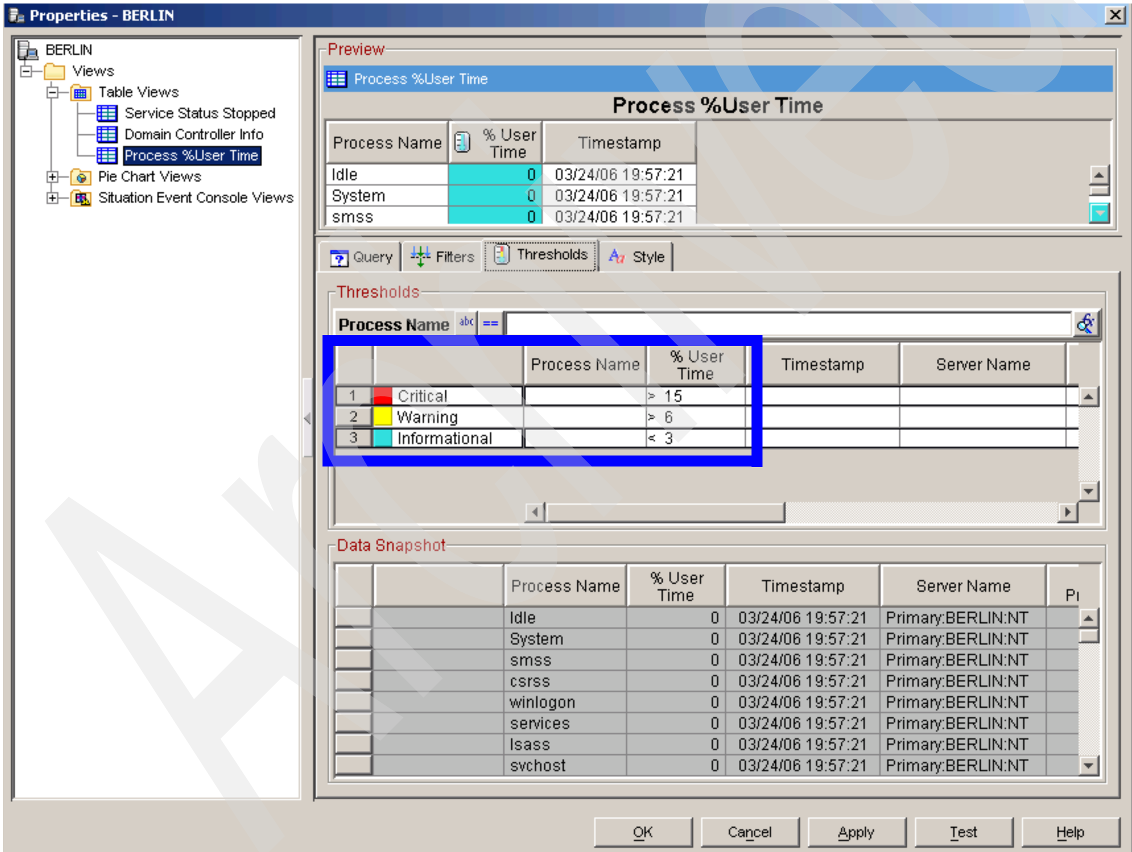


Figure 4-26 Thresholds values

9. Click the **Style** tab, select the **Show** check box, and in the Title field, type `Process %User Time`. Click **OK**. The new view is added to the workspace.

10.In Process %User Time, click **% User Time** to order the values.

You can see the thresholds working (Figure 4-27).

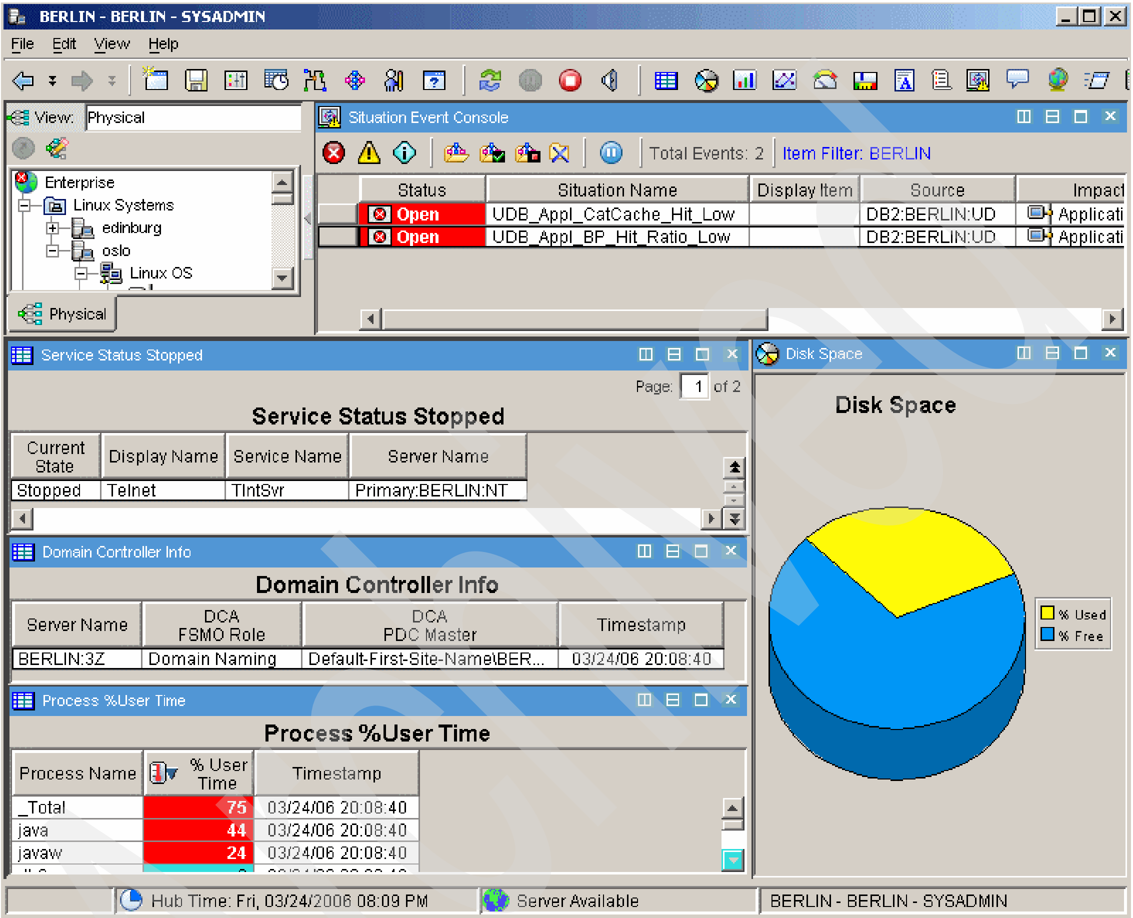


Figure 4-27 Thresholds

4.2.4 Working with a situation and events

A situation describes the conditions that you want to test on a managed system. When you start a situation, Tivoli Enterprise Portal compares the situation with the values collected by the Tivoli Enterprise Monitoring Agent and registers an event if the conditions are met. The indicator icons that appear in the Navigator alert you to the events.

This section demonstrates an example from this list and shows you how an event occurs, how a situation is triggered, and how to view the information about this event in the respective workspaces in the Tivoli Enterprise Portal client. We installed IBM HTTP Server on Windows 2000 Server in order to monitor the *Apache* process.

Each Tivoli Enterprise Monitoring Agent has a set of predefined situations that are ready to use. You can also create and customize your own situations to monitor specific conditions in your enterprise. If a situation already exists that is similar to the one you want, you can copy the original and edit the copy.

Opening the Situation Editor

You can open the Situation Editor in any one of the following ways:

- In the Tivoli Enterprise Portal toolbar, click **Situation**.
- Right-click a Navigator item and click **Situation**.
- Right-click the event item in the Navigator and click **Edit Situation**.

To launch the Situation Editor for the "NT_Missing_Msdtc_Warning" situation, perform the following steps:

1. In the Tivoli Enterprise Portal toolbar, click **Situation**.
2. Expand the **Windows Server** navigation item and click the **NT_Missing_Msdtc_Warning situation**, as shown in Figure 4-28 on page 171. Select the **Run at startup** check box.

   This option starts the evaluation immediately upon distribution, and automatically when the monitored system restarts. If you do not select this option, you have to manually start the situation on each resource.

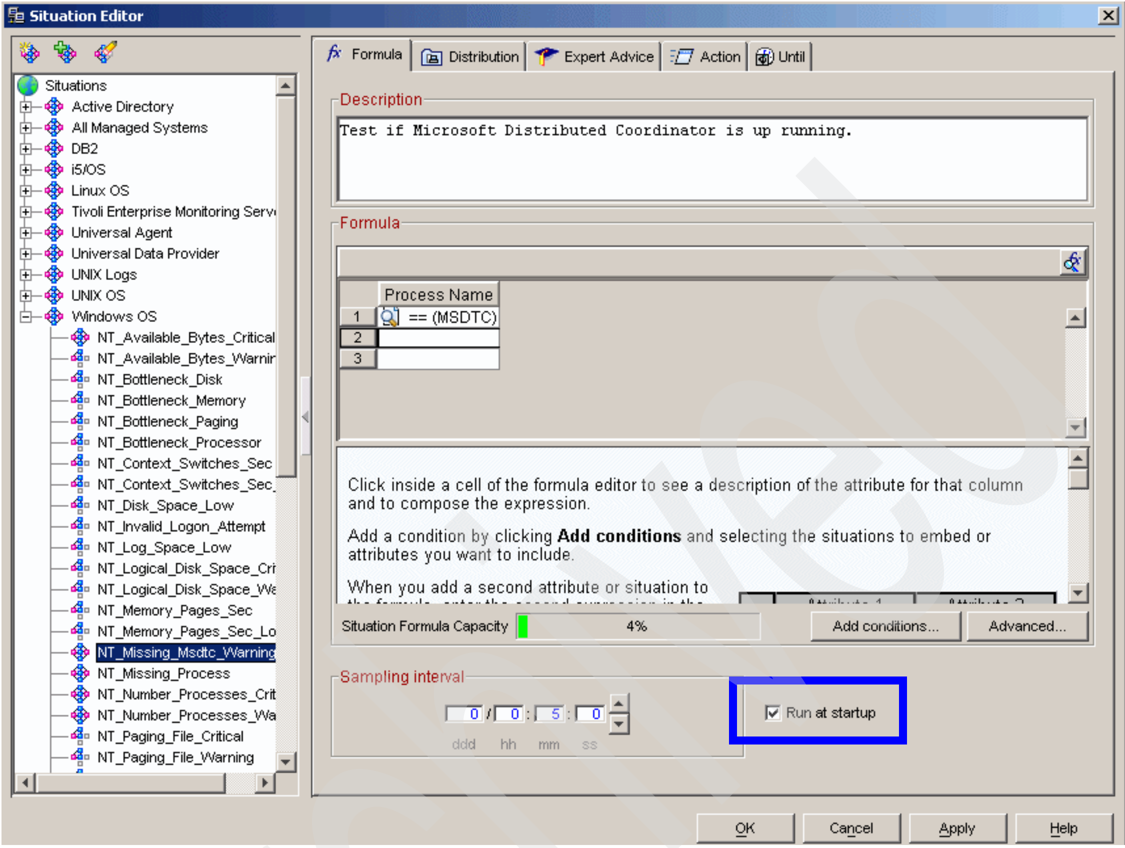


Figure 4-28 "NT_Missing_Msdtc_Warning" situation

3. Click the **Distribution** tab (Figure 4-29). You can assign this situation to a specific NT system or the NT_SYSTEM managed System List. Click **OK** and recycle the monitoring OS agent.

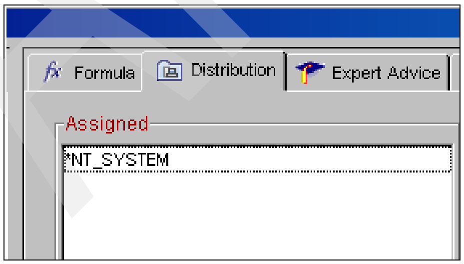


Figure 4-29 Distribution tab

4. Right-click the **Process** item in the Navigation tree (Figure 4-30).
5. Select **Situations** from the pop-up menu.
6. Expand the **Windows Server** navigation item.
7. Click the **NT_Missing_Msdtc_Warning** situation.

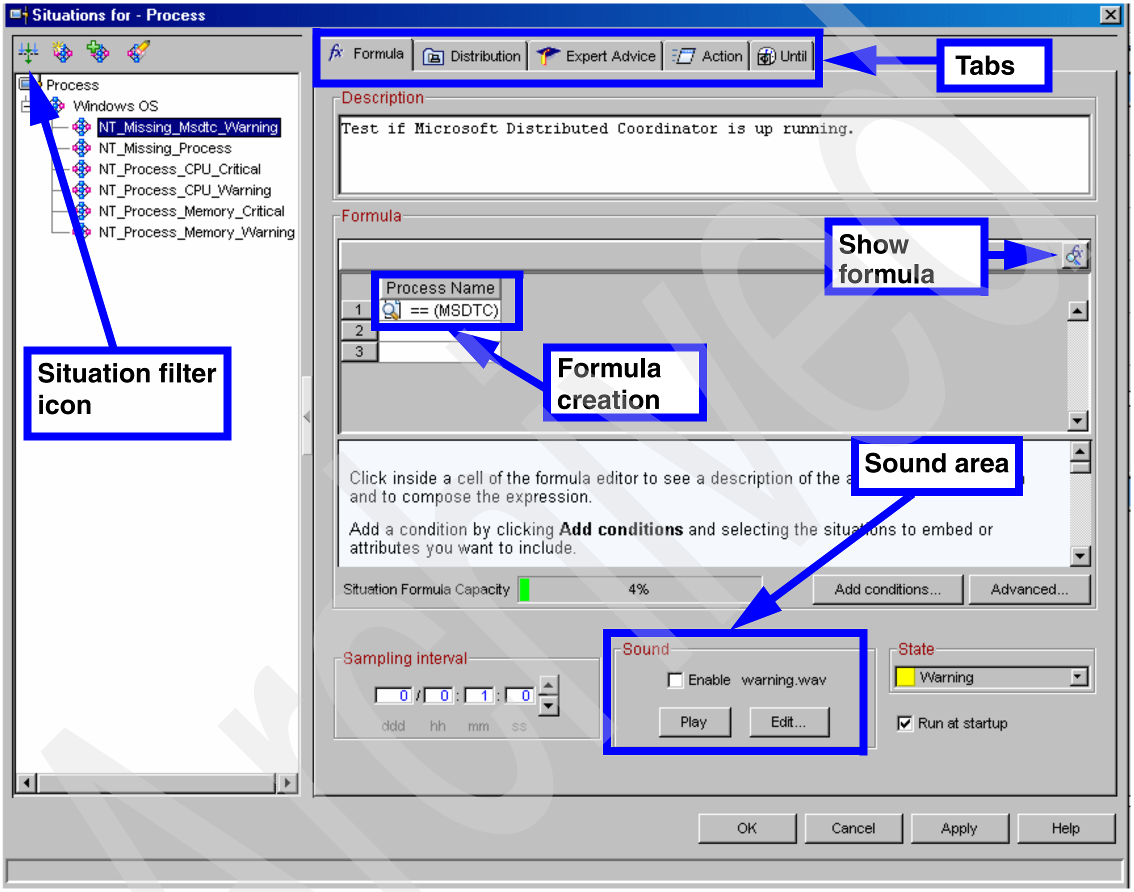


Figure 4-30 Situation Editor for NT_Missing_Msdtc_Warning

Description of the Situation Editor:

- The navigation pane shows the available situations. The default opens to the Formula tab. The other tabs across the top of the window enable you to work with other components of the situation.
- The authors complete the Description field when they create the formula.
- The formula creation field enables you to create a new formula or change an existing one.

- The Show Formula icon shows you a flowchart view of the formula.
- You can use the Sampling interval field to set an evaluation period.
- The Sound area enables the playing of a sound to accompany the alert.
- You can use the State field to identify the severity of the alert when it occurs. IBM Tivoli Monitoring Express V6.1 provides three event indicator icons. The event indicators reflect the assigned severity of the underlying system event (such as a missing process) that causes a situation to become true. See Figure 4-31.

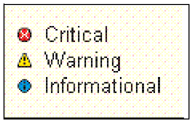


Figure 4-31 Event indicator icons

- The Run at startup check box determines whether the new situation becomes active as soon as it is distributed and applied (and automatically upon startup thereafter). If you do not select this option, you must start the situation manually each time you want to check it.

Creating a situation

This section describes how to create your own situation.

Create a situation to test whether your IBM HTTP Server program is running. The process name is *Apache*. Perform the following steps:

1. Highlight an existing situation, and click **Create another situation** (third icon), as shown in Figure 4-32. This uses the highlighted situation as the starting point.

   Alternatively, you can click the second icon to create a new situation from scratch.

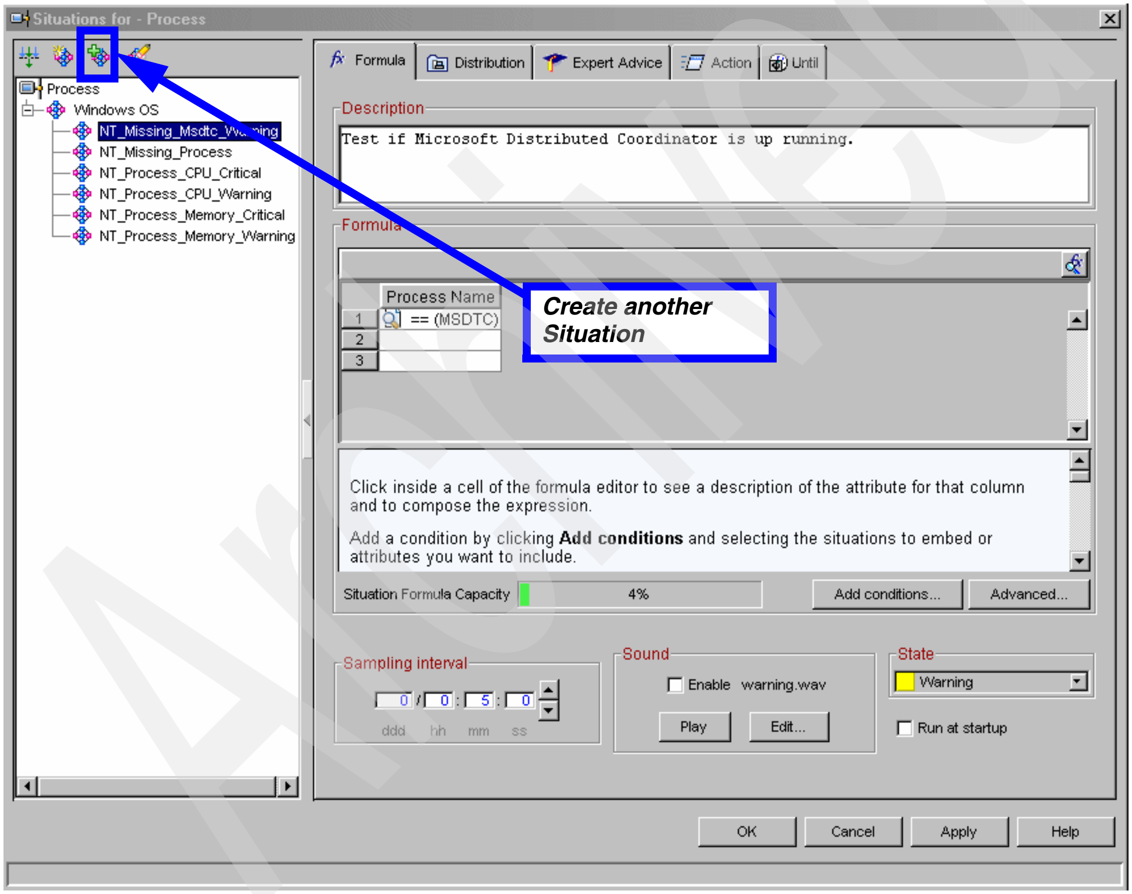


Figure 4-32 NT_Missing_Msdtc_Warning Situation Formula

2. In the Create Situation window (Figure 4-33), perform these steps:
   a. In the Name field, enter `IHS` as the name of your new situation.
   b. In the Description field, provide a description.
   c. The Monitored Application should be Windows OS (this situation uses the metrics gathered by the Windows Server Agent).
   d. Click **OK**.

   The new situation is now listed in the navigation pane of the Situation Editor.

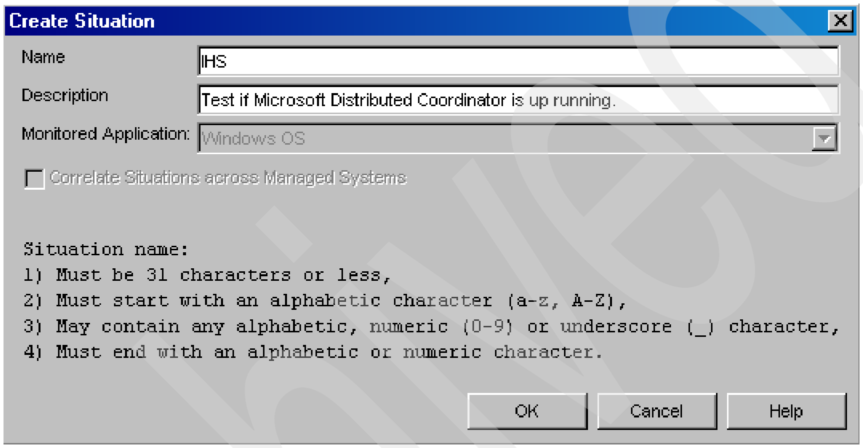


Figure 4-33 Creating a new situation called IHS

3. In the Process Name attribute column, click the field that displays **MSDTC**.
4. Change the comparison value from MSDTC to `Apache`. (It is not necessary to include the quotation marks.) By default, it starts the equation with Value of Expression Equal to (v EQ).
5. Set the Sampling interval to **30** seconds.
6. In the State field, select one of these states: Critical, Warning, or Informational.

   > **Note:** This is only an option when you create the situation in association with a Navigator item.

   In this example, we select the 

 state.

7. If you want a sound to play when this situation triggers, select **Enable critical.wav** (Figure 4-34).

> **Note:** Again, this is only an option when you create this situation in association with a Navigator item

8. Select the **Run at startup** check box (Figure 4-34). This option starts the evaluation immediately upon distribution, and automatically when the monitored system restarts. If you do not select this option, you have to manually start the situation on each resource.

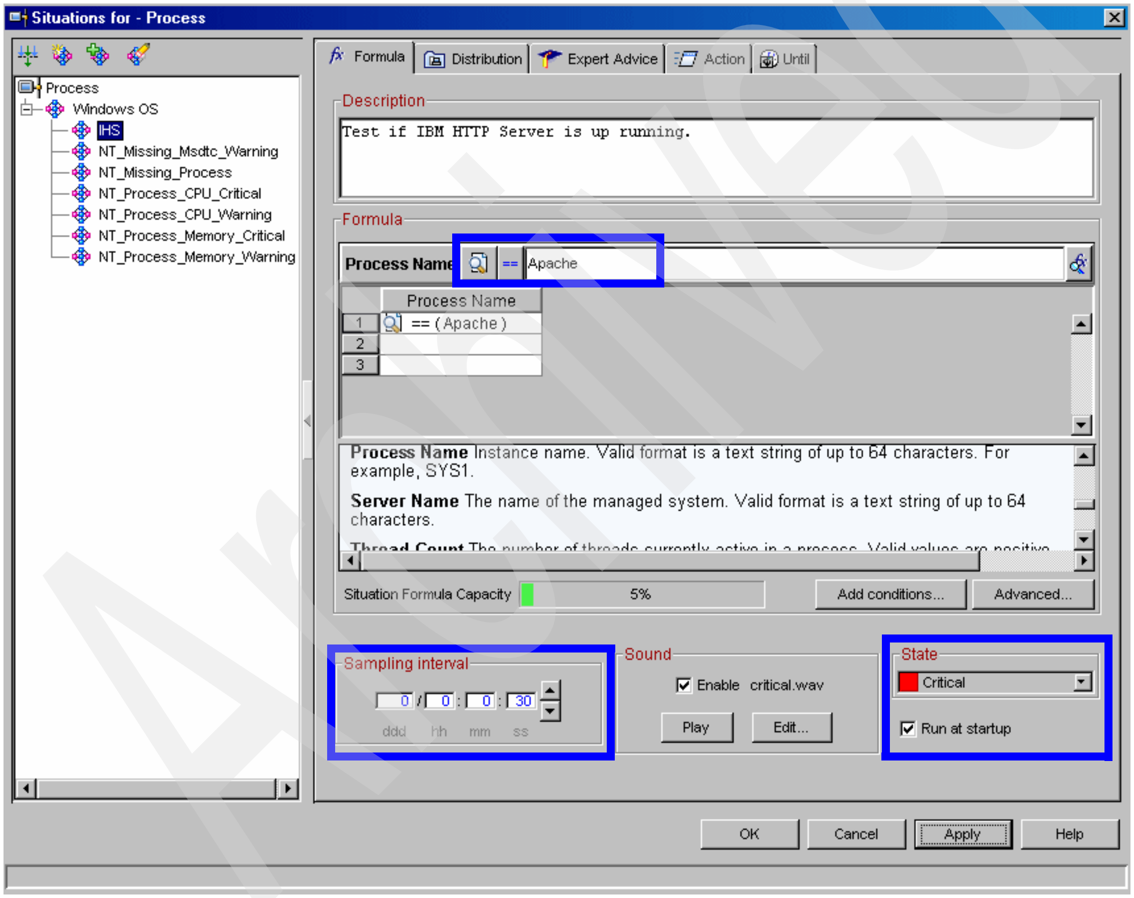


Figure 4-34 IHS Situation Formula

9. Click the **Distribution** tab. You can assign this situation to a specific NT System or the NT_SYSTEM managed System List (Figure 4-35).

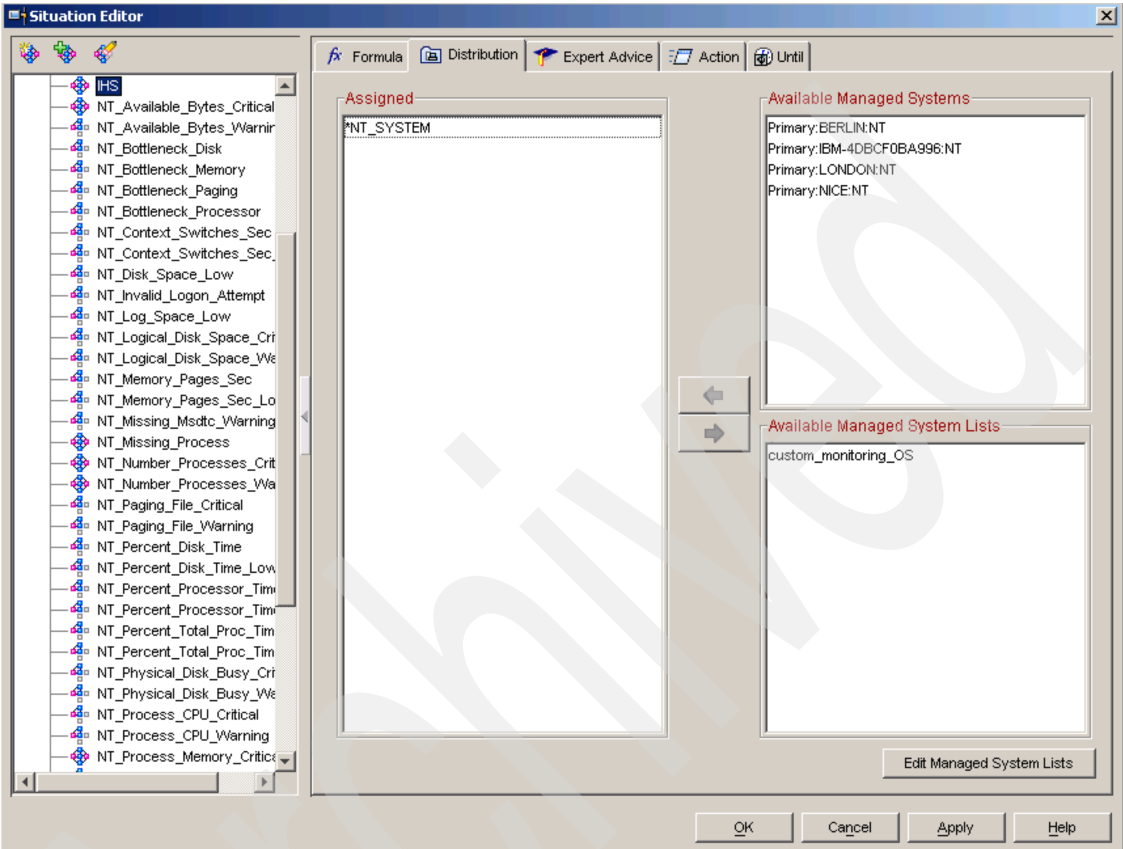


Figure 4-35 IHS Situation Distribution

10. Click the **Expert Advice** tab (Figure 4-36). Enter advice about what to do. For example, `This is expert advice for IHS. The IBM HTTP Server is not running correctly. Please notify your system administrator that we have a IHS Alert. To learn more, click here: IBM Tivoli Advice.`

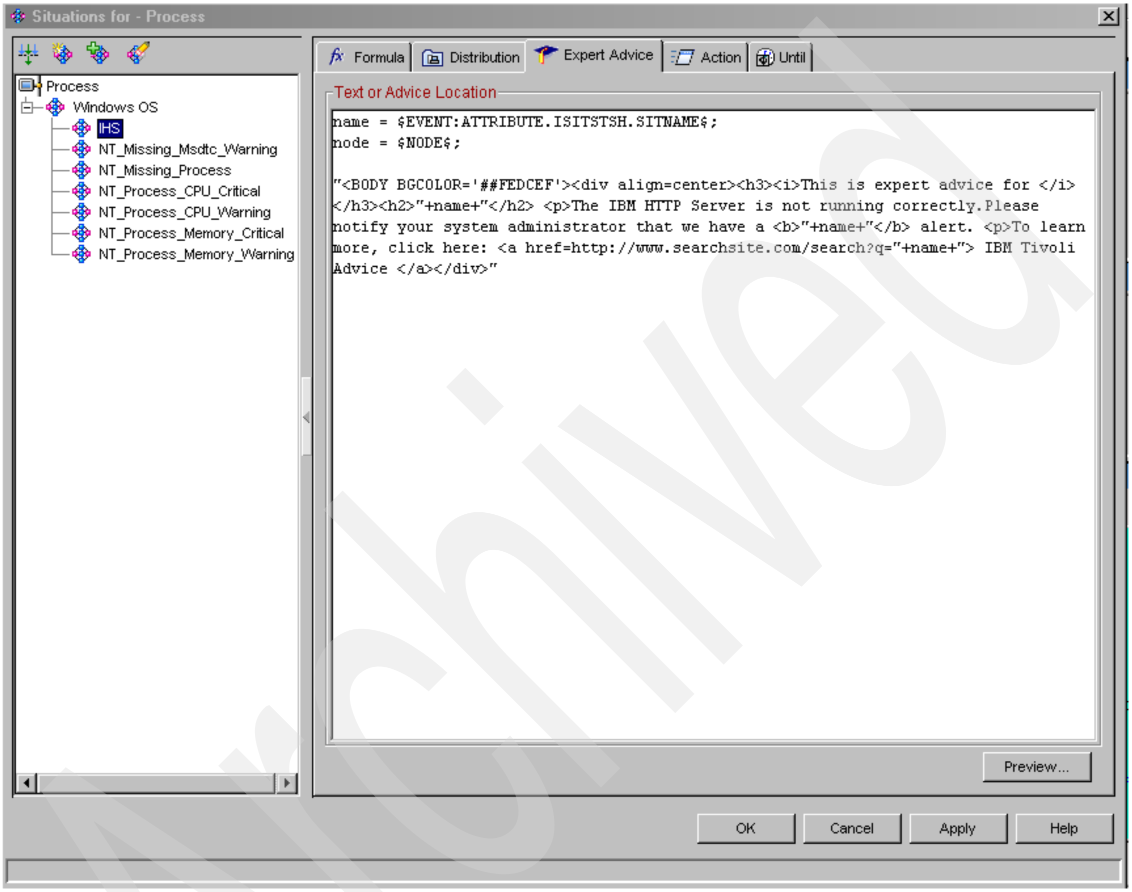


Figure 4-36 IHS Situation Expert Advice

11. To display the system command that runs when the situation fires, click the **Action** tab. In this example, we assign the following informational action (Figure 4-37):

```
echo The Process &{NT_Process.Process_Name} on the Server
&{NT_Process.Server_Name} is DOWN > IHS.txt
```

This example illustrates that you can launch a system command action to respond to the situation event result.

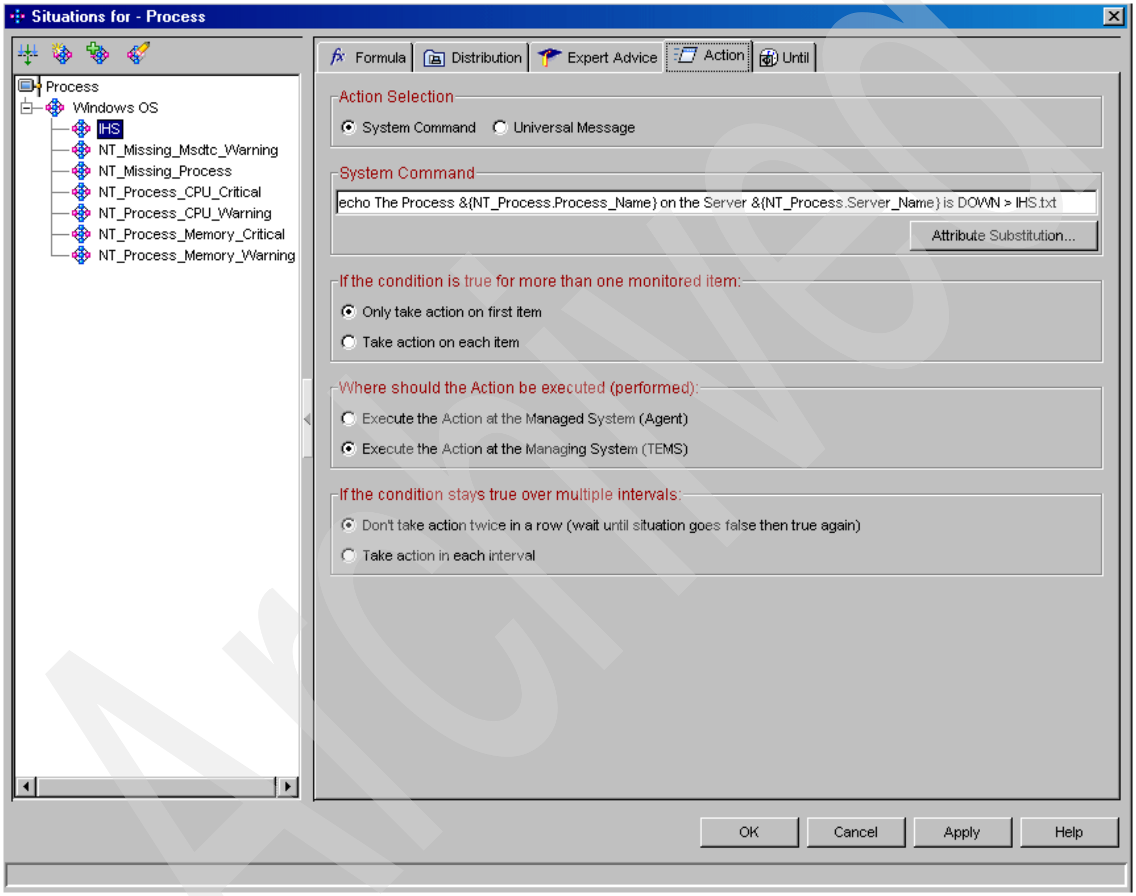


Figure 4-37 Automated action for the NT_Services_Automatic_Start situation

12. Click **OK** to save the situation.

You will now see a message in the message log on the Enterprise workspace that indicates that the situation has started.

Testing the new situation

To make the situation true, perform the following steps:

1. Stop the IBM HTTP Server application (from the main menu).
2. Watch for the alert to trigger.
3. Note that the alert appears over several Navigator items. When you hover over the red alert triangle, you see a list of all the alerts that apply to that Navigator item and below. For example, if you hover over the Windows Navigator item, you will just see the Windows event, as shown in Figure 4-38.

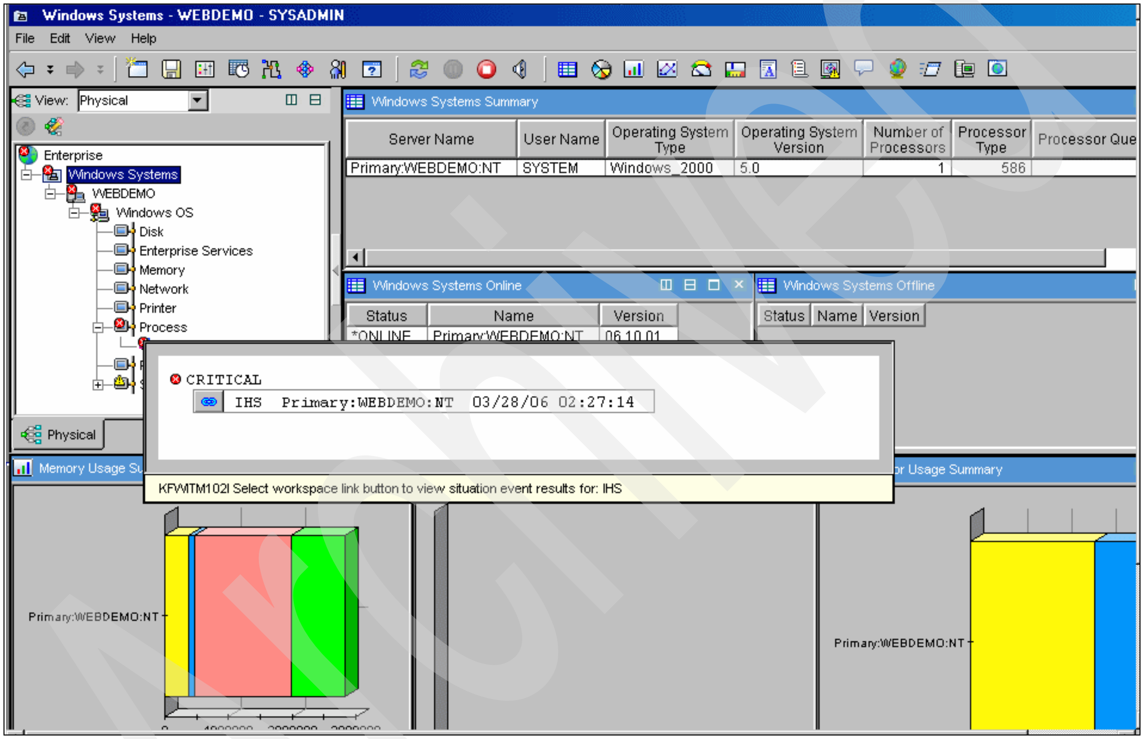


Figure 4-38 Situation event flyover

If you click the **Link** icon in event results, it takes you to a description of the event, which includes the expert advice that you entered (Figure 4-39).

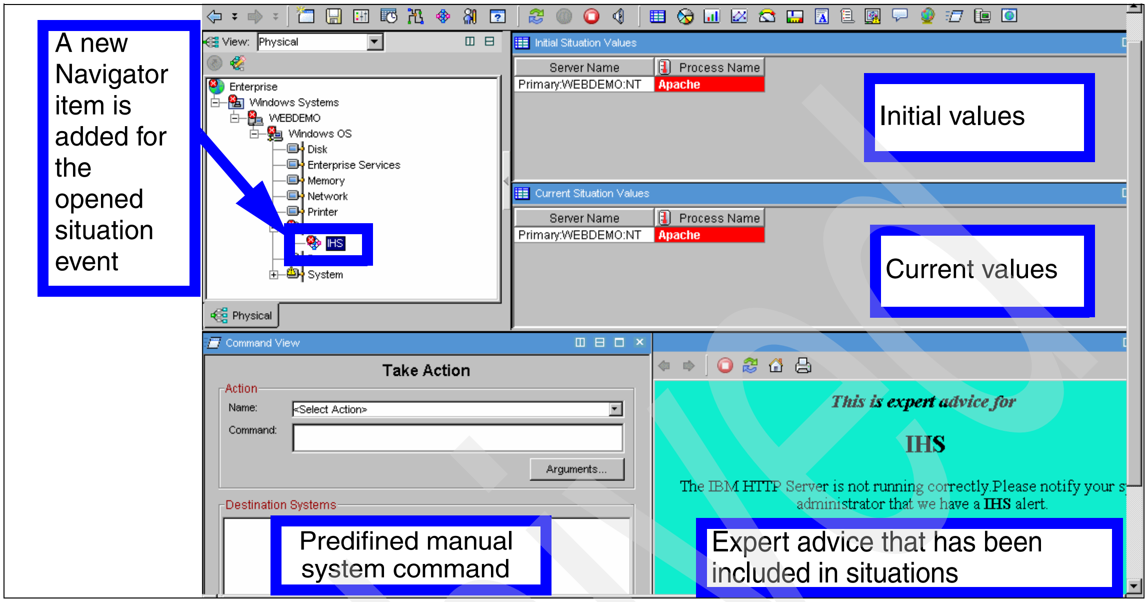


Figure 4-39 Situation event workspace

The four views in this workspace help you investigate the condition and take action if necessary:

- *Initial Situation Values* shows the values of the situation attributes when the situation fired.
- *Current Situation Values* shows the current values of the situation attributes.
- *Take Action* enables you to run a command on the managed system where the situation event occurred.
- *Expert advice* shows the information provided by the situation administrator about what to do when the situation event occurs.

4. Start your monitored application to clear the event.

Sampled events will automatically clear when the situation is no longer TRUE. Do not close them manually.

4.2.5 Launching the application IBM HTTP Server

The Tivoli Enterprise Portal user can perform another function, that is, to launch an application at a managed system. With IBM Tivoli Monitoring Express V6.1, you can perform a *launch* against a single system or a group of similar systems. Perform the following steps:

1. In the navigation panel, right-click **Windows Systems**. Select the **Launch** option in the pop-up window. The Create or Launch Definitions dialog box opens (Figure 4-40).

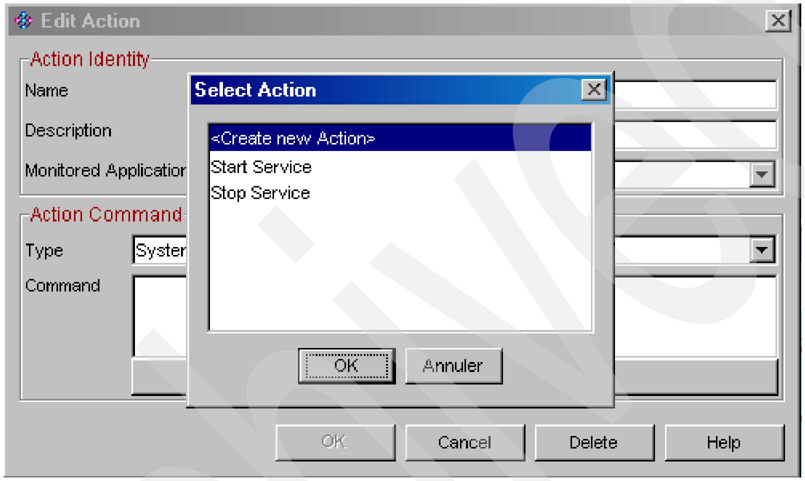


Figure 4-40 Create new Action

2. Click **Create New** to create a new launch definition. In this example, we use a BAT file to launch the IBM HTTP Server startup command (Figure 4-41).

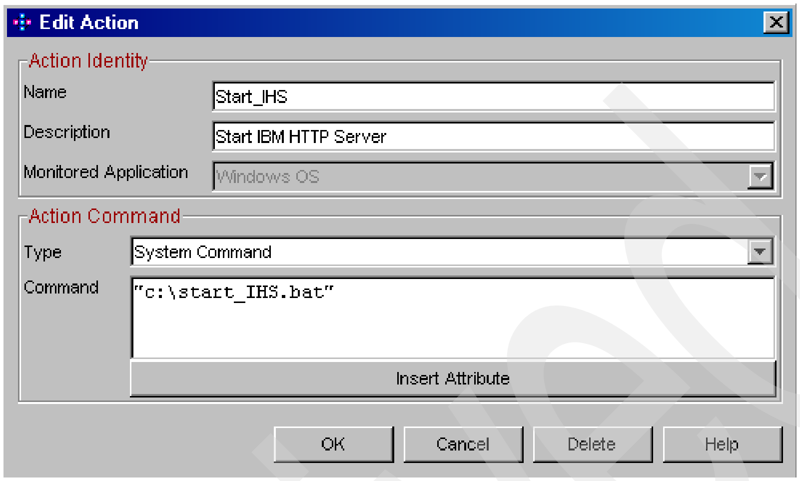


Figure 4-41 Edit Action

3. After defining a launch definition, execute the application by clicking the **Launch** button from the pop-up menu for a resource in the Navigator. The IBM HTTP Server should start on your system (Figure 4-42).

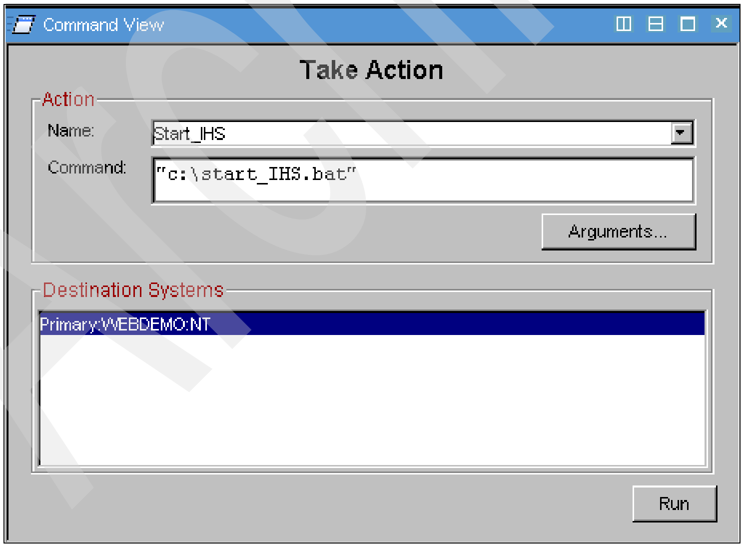


Figure 4-42 Start_IHS Action

4.2.6 Acknowledging a situation event

When you see an event indicator in the Navigator, you can create an acknowledgement. This action notifies other users that you have taken ownership of the problem that caused the event and that you are working to resolve the problem.

In this section, we show you how to create an acknowledgement for the IHS situation event that you generated earlier.

1. Right-click the IHS row in the Situation Event Console view and select **Acknowledge (**Figure 4-43).

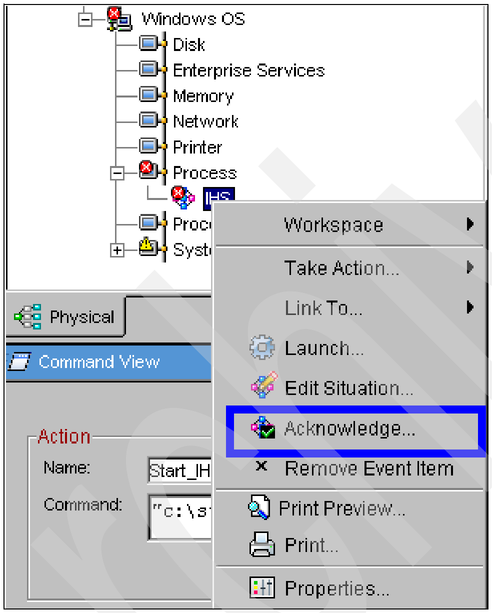


Figure 4-43 Acknowledge IHS event situation

Figure 4-44 shows the Acknowledgement window.

2. Adjust the Expiration settings so that the acknowledgement expires in 30 minutes (Figure 4-44). Adjust these settings according to your estimate of when you will finish working on the problem, for example, at the end of 30 minutes.
3. To add a time stamp to the Notes field, click the Clock symbol. Type a note that notifies other Tivoli Enterprise Portal users that you are handling the problem.
4. Click **OK** to close the Acknowledgement window.

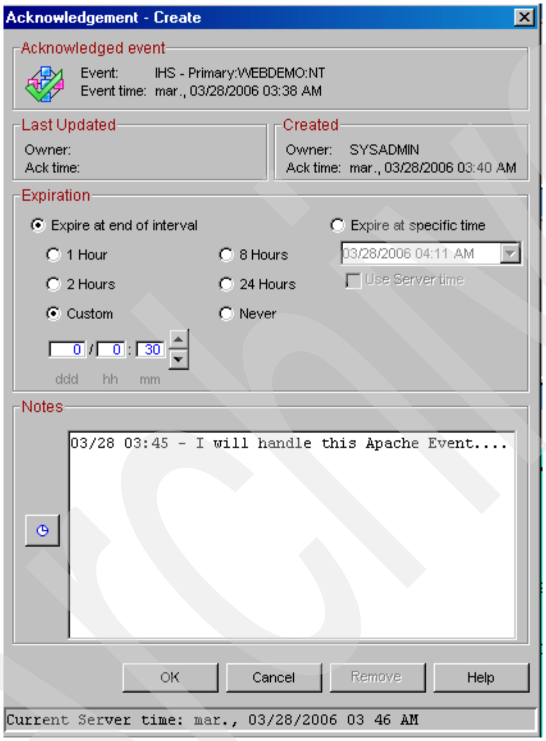


Figure 4-44 Acknowledgement

4.2.7 Working with a user profile: Creating a new user

Perform the following steps:

1. From the main icon bar, select the **Administer Users** icon (this looks like a little person with a big pencil). See Figure 4-45.

   Another way of selecting the Administer Users function is to press Ctrl+U.

2. Select the **<Default User>** entry and click the **Create Another User** icon (the icon with the little plus sign in the corner).

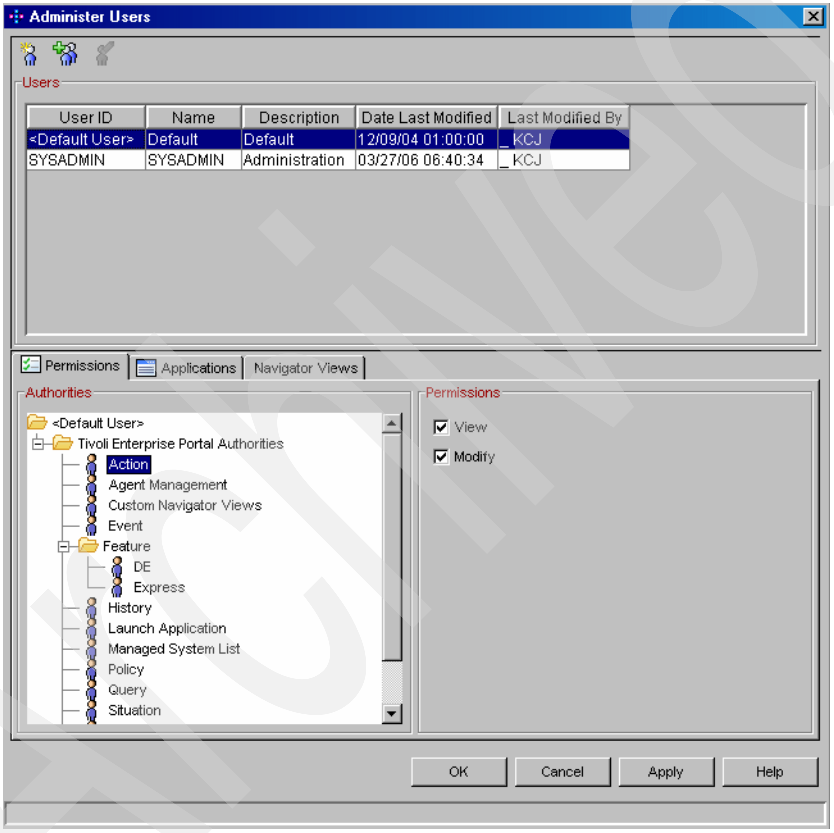


Figure 4-45 Administer Users

3. In the new window, enter `Operator` for both the User ID and User Name fields (Figure 4-46). Click **OK**.

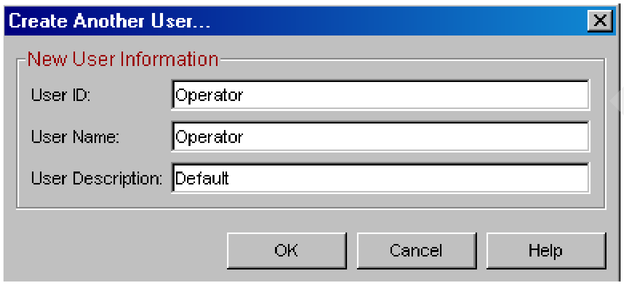


Figure 4-46 New user called Operator

You have now created a new user with the name of Operator.

4. Click the **Administer Users** icon and highlight the **SYSADMIN** entry. Click the **Workspace Administration** authority, as shown in Figure 4-47.

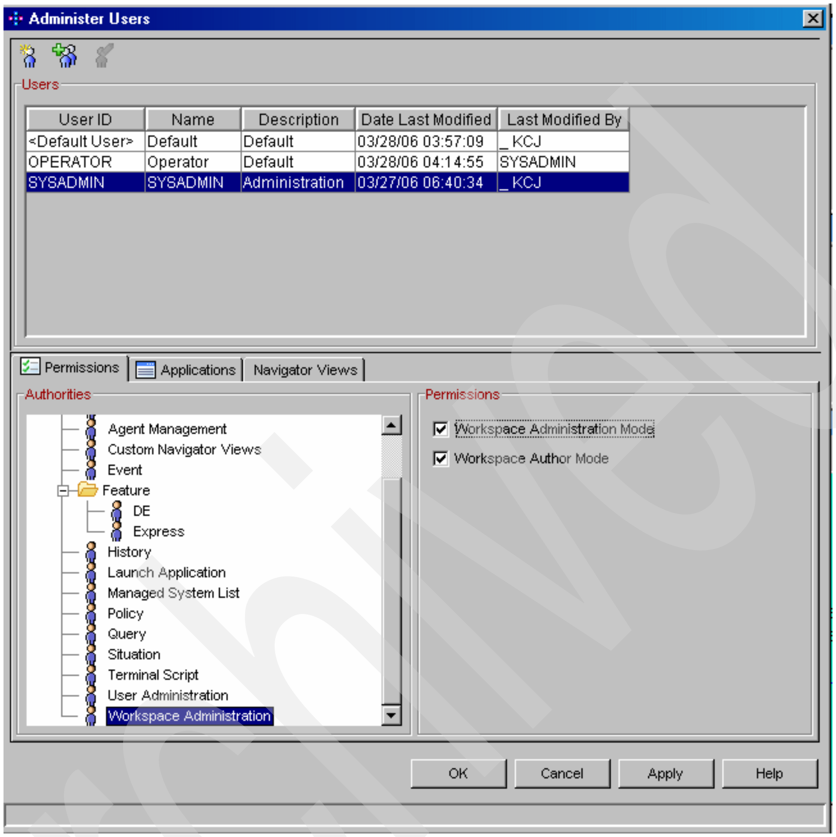


Figure 4-47 Administer Users

5. Select the **Workspace Administration Model** check box and click **OK**. This mode enables the user SYSADMIN to customize and add workspaces that are shared with all the users connected to the same monitoring and portal server.

   You now see that Tivoli Enterprise Portal shows *ADMIN MODE* in the title bar of the window.

6. From the menu bar, select **Save Workspace**.

> **Note:** We recommend that you disable the Workspace Administration Mode as soon as you finish modifying the workspaces to avoid changing any workspaces by accident.

7. Click the **Administer Users** icon again and select the **Workspace Administration** authority.
8. Clear the **Workspace Administration Model** check box and click **OK**.
9. Now close Tivoli Enterprise Portal again and log back on as the Operator user. Ensure that you can see the workspace for this user.
10. To set user permissions for specific Tivoli Enterprise Portal features, use the Permissions tab (Figure 4-48).

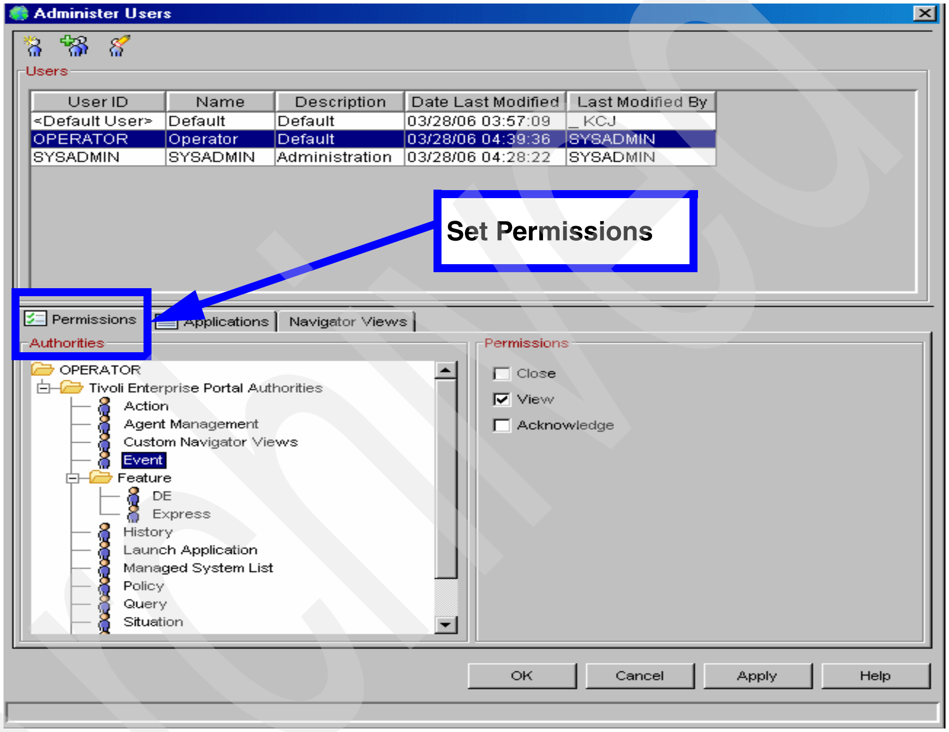


Figure 4-48 Operator Permissions

11. Now we assign applications. The user see only the areas that apply to the allowed applications. In our example, we assign the **Windows OS** application to the user Operator (Figure 4-49).

> **Note:** You have to unassign <All Applications> before you add specific ones.

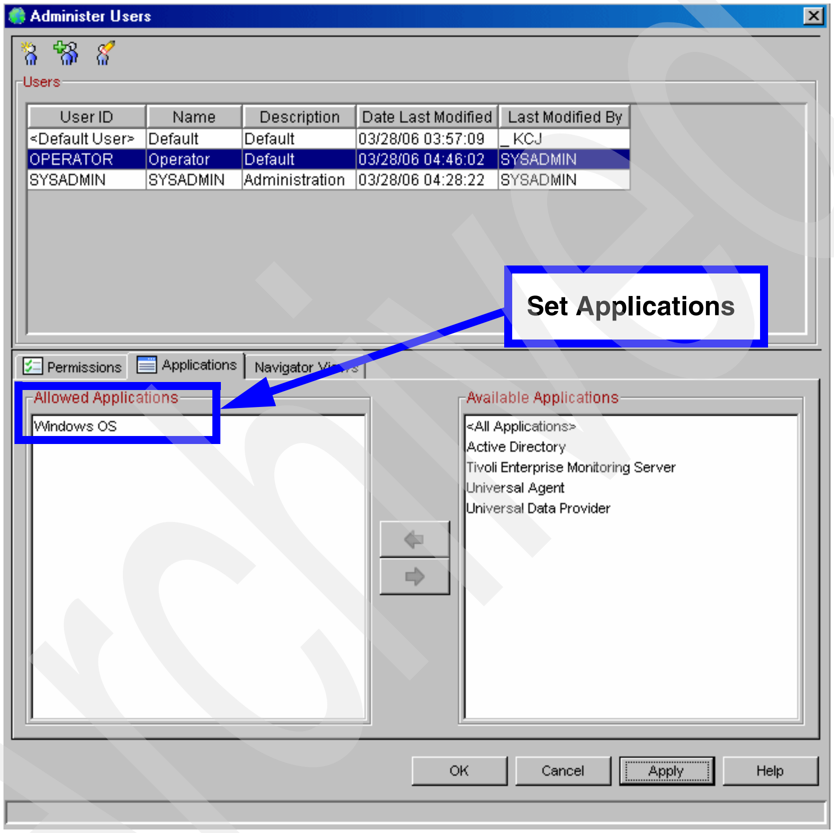


Figure 4-49 Operator Applications

12. Assign Navigator views that the user will be allowed to access, as shown in Figure 4-50.

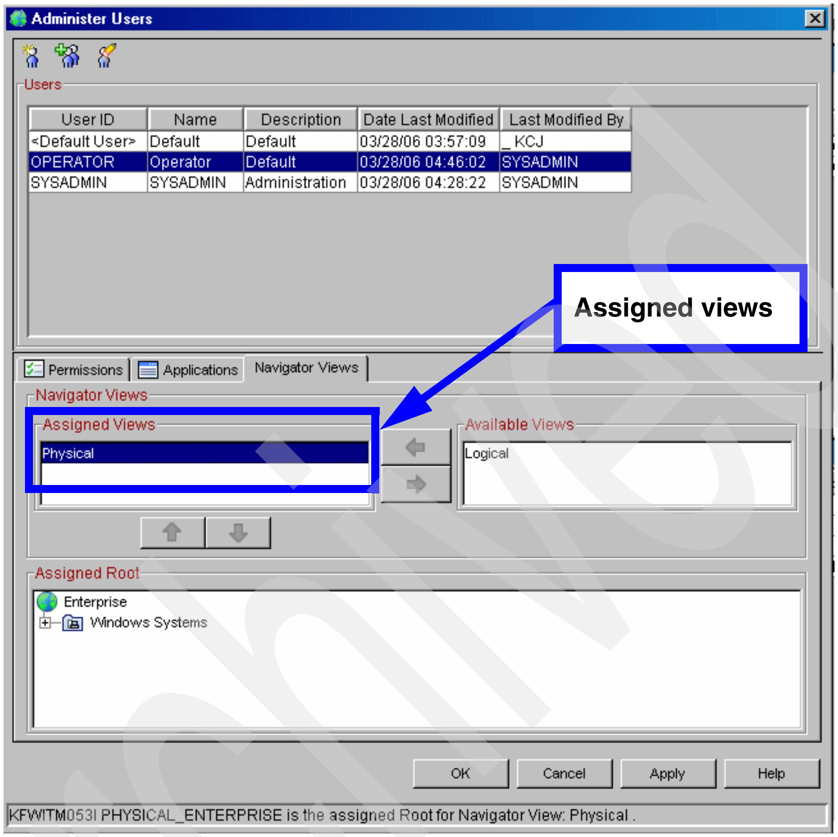


Figure 4-50 Navigator Views Operator

4.3 Historical data collection

To configure historical data collection, specify the attribute groups from which to save data samplings, the collection interval, the rollup interval, if any, and where to store the collected data. Perform the following steps to configure your historical data collection:

1. Start the Tivoli Enterprise Portal client. You can start either the browser client or the desktop client.
2. Click the icon  in the icon bar (it looks like a table with a clock over it).

3. Click the drop-down menu and select the **Windows OS** option. This gives you a list of all the different attribute groups that are tracked for a Windows machine (Figure 4-51).

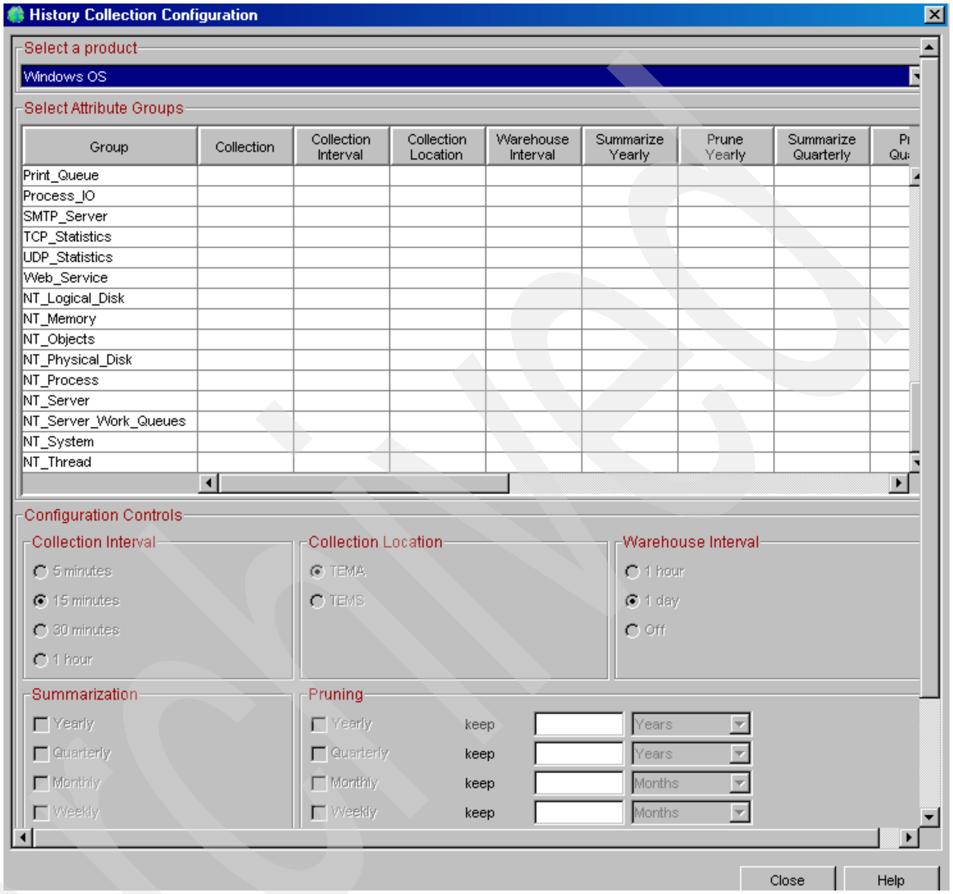


Figure 4-51 Historical Collection Configuration

4. In the Collection Interval field, select **5 minutes**. This way you can see the results within the next 5 minutes. In a production system, the user can typically select 1 hour or 15 minutes depending on the granularity of the data desired.

5. In the Collection Location field, select **TEMA**. This collects 24 hours' worth of data on the endpoint in a rolling log. The user can also opt to collect the data at the Tivoli Enterprise Monitoring Server, but we do not recommend this.

6. In the Warehouse Interval field, select **1 hour**. This means that the data from the endpoints is collected every hour to be rolled into Tivoli Data Warehouse. The other option is to do this on a daily basis. The decision depends on the criticality of the monitoring data and performance considerations. The 1 hour selection ensures that you will see the database tables being created and the data being entered into the warehouse at the top of the next full hour.
7. In the Summarization field, select **Monthly**. This automatically selects all the items underneath. You can clear any of these if you are not interested in these summarizations. For example, you might be interested in aggregating data on an hourly, daily, and monthly basis only and can forego the weekly summing up of data. In this example, keep them all selected.
8. In the Pruning section, select the following items and specify the corresponding values for data retention:
   - Monthly: 36 months
   - Weekly: 12 months
   - Daily: 3 months
   - Hourly: 30 days
   - Detailed data: 14 days

9. Click the **Configure Groups** button to make these settings apply to the selection you made previously (Figure 4-52).
10. To start data collection for one or more attribute groups, specifically perform the **Start Collection** action. Check for the correct operation of the data collection. After a minute or two, make sure that the Windows OS agent is still running.

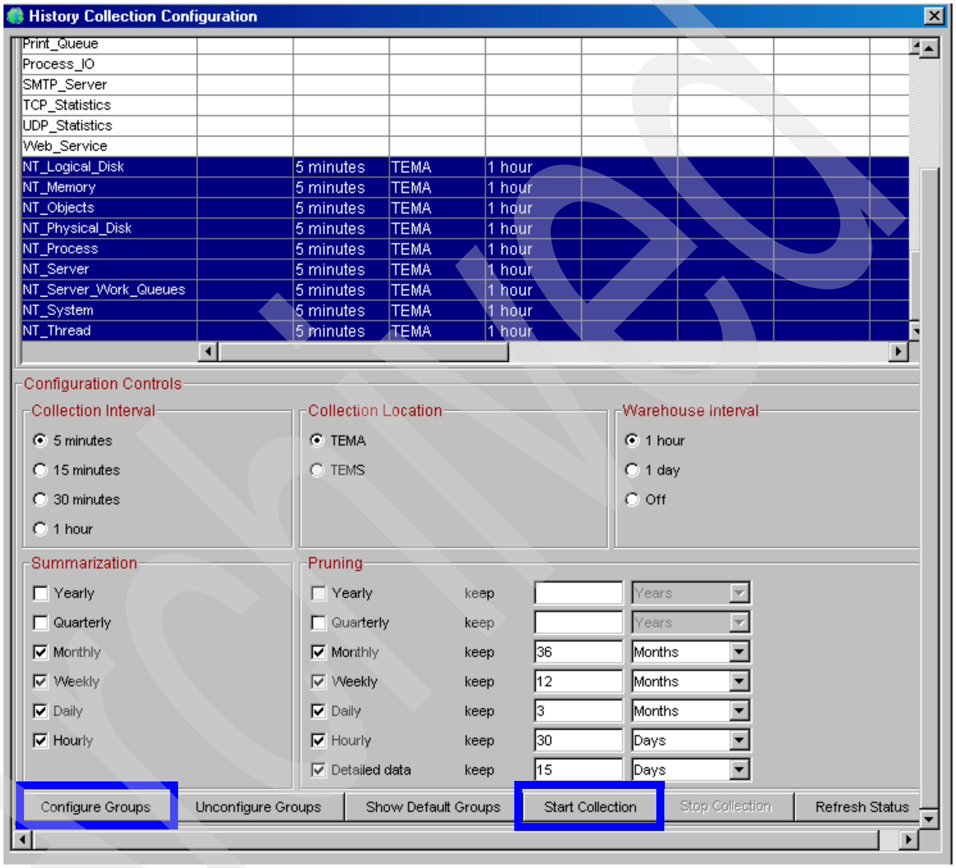


Figure 4-52 Data Collection Configuration

Displaying historical data in Tivoli Enterprise Portal

When you now look at any view in Tivoli Enterprise Portal, which displays data for an attribute group that you have selected for historical data collection, you will see an additional icon at the top left of the graph. It looks like a little window with a question mark and a clock.

Perform the following steps:

1. Click the **Memory** workspace in the Windows OS physical tree view (Figure 4-53). You see that all the three graphs show this icon.

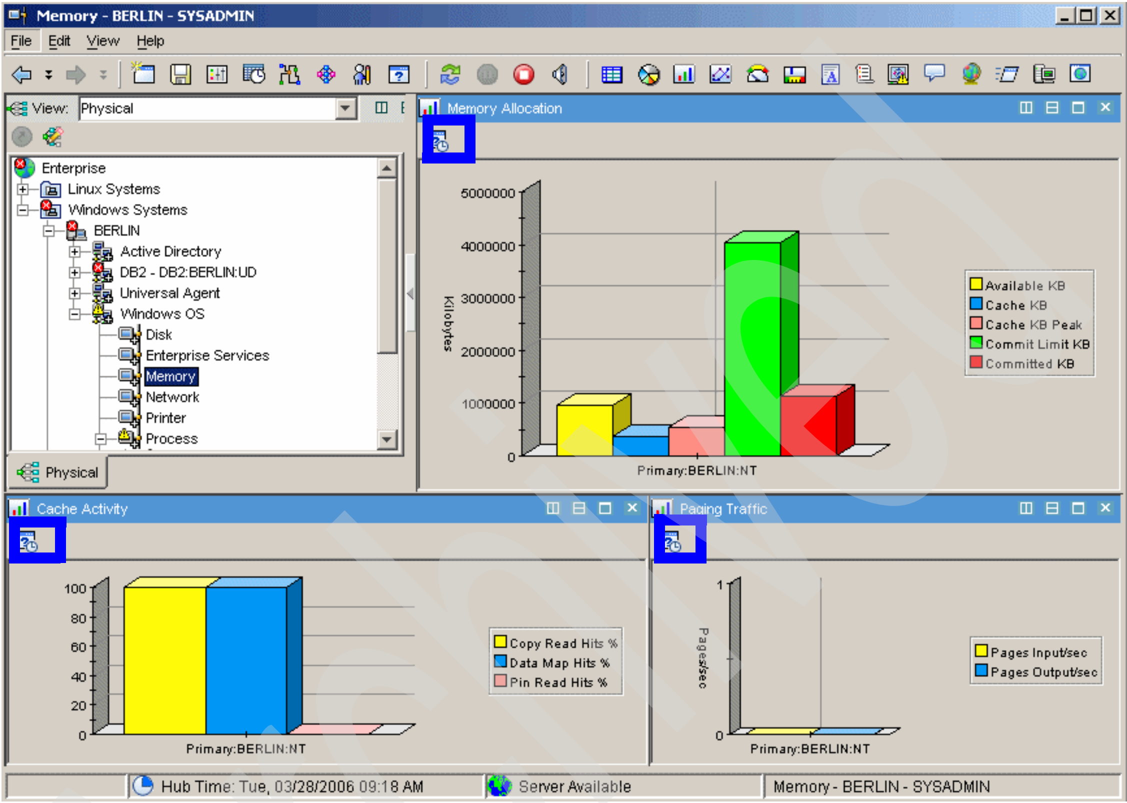


Figure 4-53 Memory workspace window

2. To select historical data in the Memory Allocation view, click the icon.

3. This opens the Select the Time Span window. The default setting is Real time. This means that real-time data is displayed in the graph. So every time you refresh the workspace view, the current data from the agent is displayed.

   However, we want to look at the historical data. Therefore, select the **Last** radio button and specify **1 Hours** to look at data for the last one hour, as shown in Figure 4-54. Click **OK**.

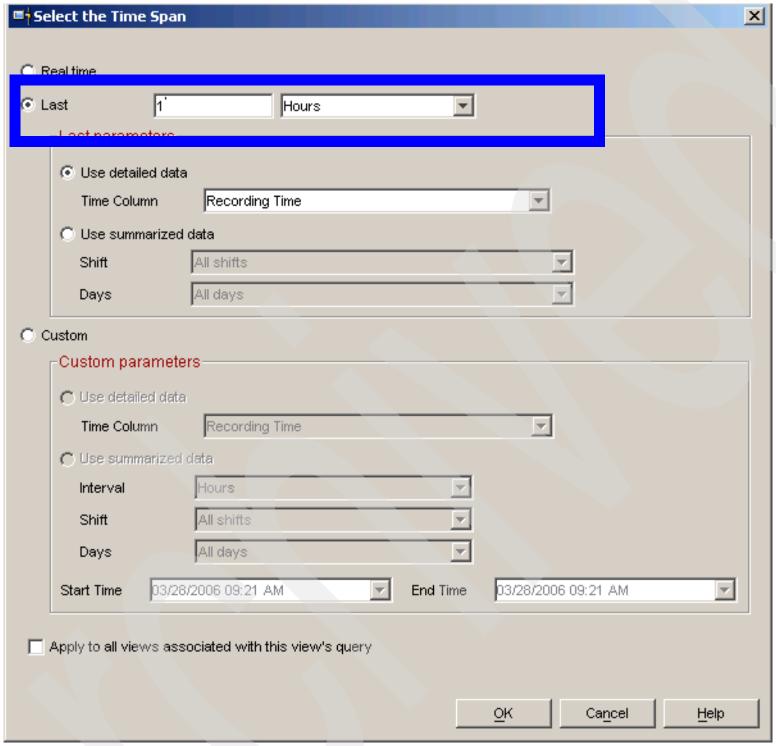


Figure 4-54 Time Span

You see that the graph (Figure 4-55) immediately updates to show the last hour of data (or at least the portion of the hour for which it has collected the data, because it is unlikely to have collected a full hour's data by now).

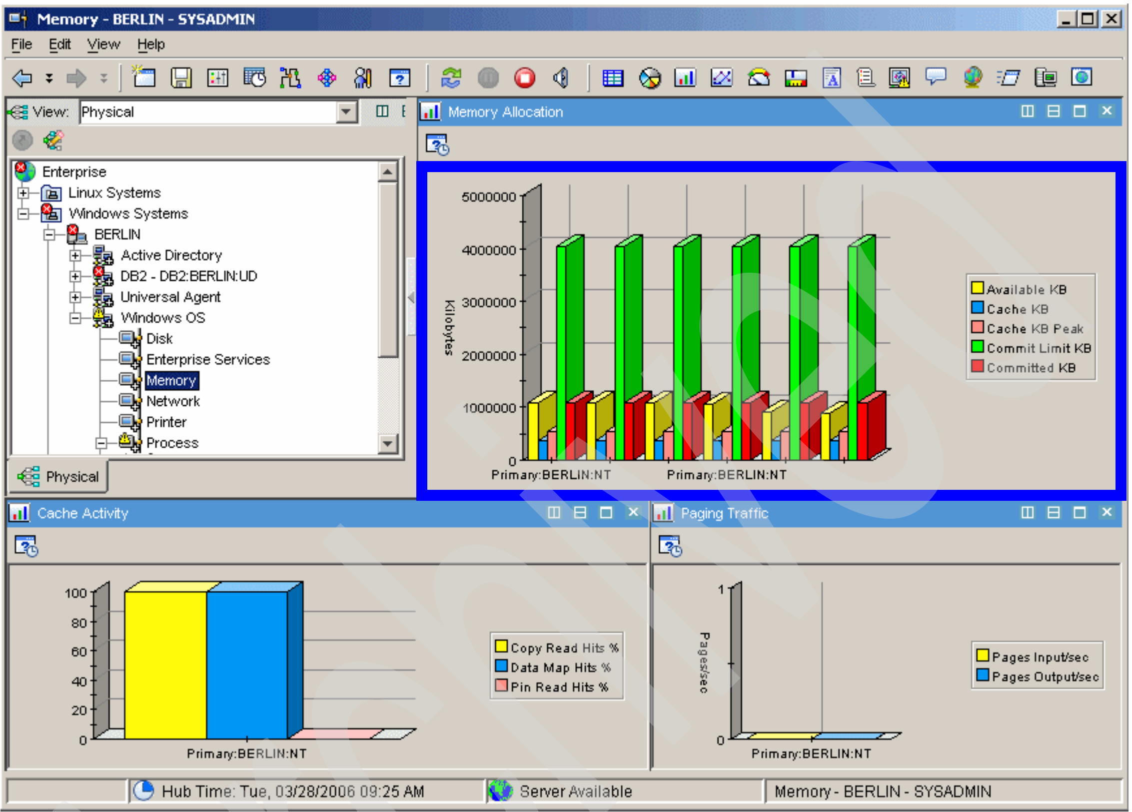


Figure 4-55 Memory Allocation historical data

4.4 Solution Installer tool

The Solution Installer is an InstallShield MP-based installer that provides an interface for installing the built solution into an existing Tivoli Enterprise Monitoring Server, Tivoli Enterprise Monitoring Server depot, Tivoli Enterprise Portal Server, or Tivoli Enterprise Portal installation. Using the Solution Installer tool, you can import some or all of custom workspaces into your Tivoli Monitoring Express environment.

Important: This tool is available at the IBM Tivoli Open Process Automation Library for Business Partners Web site:

http://www-18.lotus.com/wps/portal/tm

You can import the following custom workspaces:

- kud_sbeDBs_solution: Contains custom workspace files for IBM DB2 UDB.
- koq_sbeMSSQL_solution: Contains custom workspace files for Microsoft SQL Server.
- kum_sbeWAS_solution: Contains custom workspace files for IBM WebSphere Application Server.
- k3z_sbeActDir_solution: Contains custom workspace files for Microsoft Active Directory Server.
- knt_sbeWin_solution: Contains custom workspace files for the Windows operating system.
- klz_sbeLnx_solution: Contains custom workspace files for the Linux operating system.

Important: Before importing custom workspaces for Universal Agent solutions, make sure that you have imported the metafile and scripts required for both the IBM HTTP Server and the IBM WebSphere Application Server Universal Agent solutions on the server that is monitored.

5

Sample scenarios

This chapter provides a blueprint for turning a monitoring need into an enterprise event, along with an in-depth look at the features that are available and that enable monitoring beyond single platforms or components. These features include the Navigator, queries, filters, Take Action, situation actions, and situations.

In this chapter, we discuss the following topics:

- Owning the tool
- Defining the need for monitoring
- Understanding the terms
- Building the monitor
- Building a monitoring view
- Monitoring the Tivoli environment
- DB2 UDB scenarios
- Windows Active Directory scenarios
- Apache on Linux scenarios
- Microsoft Internet Information Services scenarios
- Microsoft SQL Server scenarios
- IBM Tivoli Universal Agent scenarios
- SOAP scenarios

Before discussing the owner of the tool, defining the needs, and so on, a few words about how this chapter is constructed.

Most of the monitoring products available on the market today provide the capability to monitor your environment with minimal configuration. IBM Tivoli Monitoring Express V6.1 is no different.

With IBM Tivoli Monitoring Express V6.1, you get the following capabilities without having to customize the configuration:

- Monitoring rules (situations). We discuss this in 5.4, “Building the monitor” on page 212.
- Monitoring views (workspaces). We discuss this in 5.5, “Building a monitoring view” on page 234.

Because we already discussed historical data collection and reporting d in the previous chapters, we do not cover these topics here.

After the introduction, this chapter provides details about creating new monitoring rules that best suit your environment, associating them to customized Navigator views, and creating your own customized monitoring views.

We provide simple case studies describing how a mid-market company can manage the health and availability of an IT infrastructure. These case studies examine the creation and deployment of monitoring solutions that enable a mid-market company to visualize the computing resources in their IT infrastructure in order to monitor and react to any event that might affect the delivery of critical business services. The case studies focus on understanding how to identify what common resources should be monitored, how they should be monitored, and common corrective actions that can be used to respond to situations that occur. In the process of understanding these monitoring requirements, this solution illustrates how to build a monitoring console that can be easily used by support personnel

At the end of this chapter, we provide information about how you can integrate other vendor products with IBM Tivoli Monitoring Express V6.1 using the Universal Agent, and how you can use the SOAP command to further enhance and integrate information with IBM Tivoli Monitoring Express V6.1.

5.1 Owning the tool

Before discussing the real-life application of the tool, a basic question should be addressed. Who will own the tool?

An organization performing enterprise systems management can be thought of as being somewhere along an evolutionary process for distributed computing. Although the host arena is mature, the distributed environments are generally less mature. This is not to say that a less mature environment is 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, you can expect the Tivoli support team to own the monitoring solution and the development of monitors for the enterprise.

An increasingly common movement among more mature enterprise customers is to turn their developed products into services. In such situations, the Tivoli support team manages the infrastructure, sets the direction, and provides training for the use of the services, in addition to providing support to those using the services. In this context, for IBM Tivoli Monitoring Express V6.1, creating monitors is left to the administrators and the application team, while the overall tool function and architecture is held by the Tivoli support team.

To determine how to manage an implementation, use the following criteria:

- IBM Tivoli Monitoring Express V6.1 is already installed in the environment and is stable. This includes the implementation processes. The organization's Tivoli team has acquired skills pertaining to IBM Tivoli Monitoring Express V6.1.
- The administrators and the application team have the skills required for programming, that is, the general skills.
- You are prepared to educate the administrators and the application team about how to build the monitors and provide documentation to support this.
- Precedence exists for the administrators and the application teams to "own" their own monitors, *and* they are ready to "own" their monitors, that is, recognizing the resource requirements within their teams.
- The change control to production environment is a fairly mature process, and the application team and administrators follow the process.
- Development and quality assurance (QA) environments are available for the application teams and administrators to develop their monitors.
- A commitment to a QA process will be made prior to the introduction of the monitors into the production environment.

If your organization is not in a position to meet these criteria, it will be better served by holding control and developing and managing the tool in a single group.

5.2 Defining the need for monitoring

In this section, we discuss the methodology to turn a monitoring need into a technical solution answering that need. We describe non-technical issues, that is, those relating to process, in addition to the technical details.

Five distinct types of activities are advocated for defining the need for monitoring:

- Identifying a monitoring need. Here we discuss how the nomination should come from either a business problem or from the problem management activity within your organization.
- Identifying the target audience. This topic describes the importance of ownership with regard to the events. The development of monitoring should always be carried out in coordination with the system and application administrators and the help desk.
- Identifying and refining the possible events list. This topic provides details about coming up with a complete solution by investigating an event request.
- Meeting the target audience for approval. This is a very important activity. At this point, all the parties concerned should agree to an escalation process for this event if changes are required.
- Creating, testing, and implementing the monitor. Here, we discuss the processes 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 use of the tool in your environment (enterprise impact).

5.3 Understanding the terms

It is essential that you understand the following terms as defined in *IBM Tivoli Monitoring, Version 6.1 User's Guide,* SC32-9409:

Event An action or an 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 is displayed 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, and the default is 15 minutes.

Pure event A pure event, such as a paper-out condition on the printer or writing a new log entry, occurs automatically. Situations written to notify 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 occur 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. Alternately, it can be closed 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, that is, whether critical, warning, or informational. It is indicated by a colored event indicator. The 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 window or frame in a workspace. It might contain data from an agent in a chart or a table, or it might contain a terminal session or browser. A view can be split into two separate, autonomous views.

5.3.1 Pure versus sampled events impacting the IBM Tivoli Monitoring Express V6.1 console user

A pure event can be understood to mean a stateless event. There is no recognition of the state of a resource in a pure event, for example, when an event is read from a Microsoft Windows log file.

In the case of the log file monitor, the agent is merely matching lines from the log to which it has been configured for forwarding to the Tivoli Enterprise Monitoring Server. The agent is not keeping track of anything to compare events, and there is no evaluation other than matching.

There is no concept of an interval when building a situation to detect pure events, although some configuration of a time interval is possible for most agents that detect pure events. However, this pertains to all the agent operations and not to a situation-by-situation basis, as is the case with sampled events.

In contrast to a pure event, a sampled event has a state. The current state of a 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, you evaluate the current status of the storage application process for up/down, it would be a sampled event.

You can evaluate the status and compare it against some criteria, for example, up/down. When the monitor determines that the criteria have been met, the sampled situation becomes true and appears on the event console. When it is resolved, it is false.

When a pure event comes to the console, it is there until acted on by a human operator, that is, if you are managing the events from the console, unless you include an Until setting to expire them at a later time. For more details about Until, refer to 5.4.7, “Using the Until tab” on page 226.

A small business support team (or any organization without an enterprise view tool) using the events console needs 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 5-1 illustrates such a console.

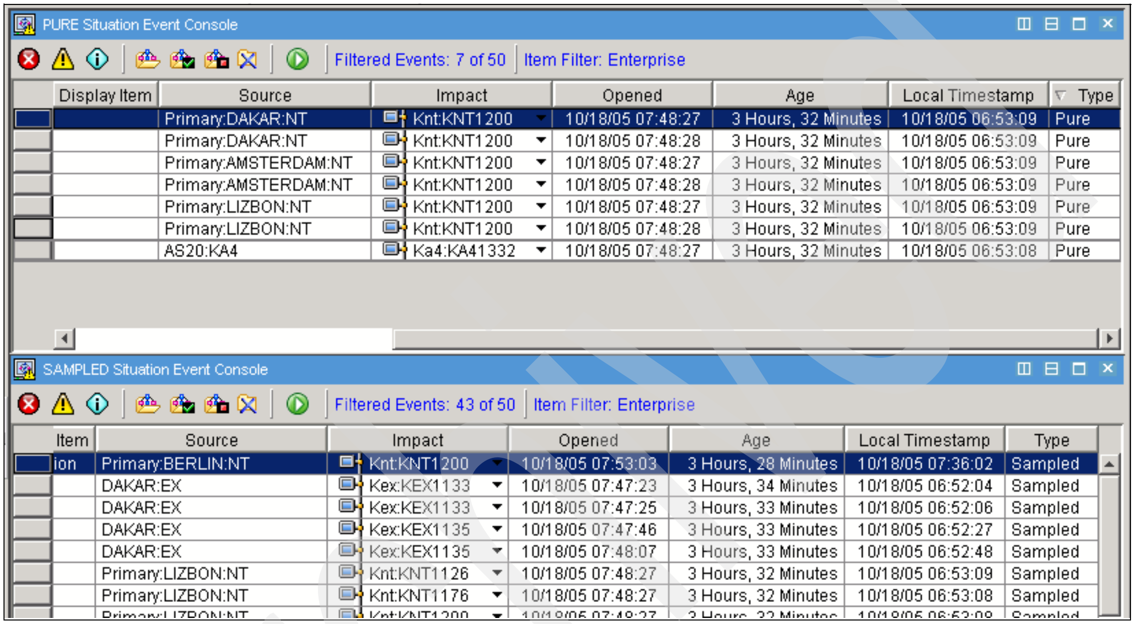


Figure 5-1 A custom workspace with separate pure and sampled event views

5.3.2 Pure versus sampled events impacting the customer

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 receive 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 can vary from agent to agent.

What to do with these events is easily recognized and fits within our traditional notion of enterprise monitoring.

For customers who use the IBM Tivoli Monitoring Express V6.1 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 can 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 a person arrives on the scene, the sampled event might 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, for example.

These steps outline the suggested best practice for dealing with sampled events:

1. When the sampled situation becomes true, attempt to resolve the issue through 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. If the situation has not become false after the next iteration, you will want to take action (such as generate trouble ticket) or execute a command that might fix the problem.

If automation is not possible, you might want to inform the person noticing the event about what to do next so that a resolution process can be started as soon as possible (according to agreed response times negotiated in your service level agreements).

5.3.3 Identifying a monitoring need

This identification includes recognizing the need for application monitoring for new applications, improving the existing monitoring solutions through the introduction of newer, more sophisticated monitoring, or correlation among existing and known monitoring items. This should include identification of the criticality of the request and the impact of the situation 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 Information Technology Infrastructure Library (ITIL) concept of the problem management function within your organization.

When a situation occurs that is of enough significance in your organization and that can be detected through some technical means, the problem management function should request through 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; observe these processes in the monitoring space and recognize them as part of this activity.

The second way in which the need for monitoring is discovered is through the requirement for application management. All newly developed and purchased applications in the environment should be monitored as part of systems management to the level deemed necessary according to the importance of the system or application.

In the case of off-the-shelf software, the vendor should be able to provide error conditions for trapping, along with advice about which conditions are most critical. For developed software, the developer should build the software with monitoring in mind. This means that issuing alerts about possible problems prior to failure should be a function of the application.

Monitoring that is merely reactive, that is, identifying problems post-failure, can never be proactive in assisting the administrators of the application in preventing failure. Process-down monitoring is important, but what is far more valuable is a set of alerts that help recognize that conditions that might precipitate or precede a failure exist.

Expanding on this concept, would you rather somebody warned you about an approaching tornado or would you want to wait for it to hit your house to react? As sure as you are about your answer to this question, it is very common for administrators to say “Do not bother me until it actually goes down.” Although 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, go through a planning session before designing your monitoring and dashboard views. We recommend the following approach:

- Define the objectives of the solution.
- Ensure that business objectives are stated.
- Define the scope of the solution.
- Identify the most critical applications.

Examine the existing technology infrastructure to see which needs can be met with the existing solution. If there are some aspects that 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 in the team that develops or implements the monitoring solution. This role can be performed by the manager of the team or a business analyst within the team.

The “gatekeeper” should use the criteria described at the beginning of this section to make sure that the resources that are required for building, maintaining, and operating the monitoring solution are appropriately managed by the current request, as also are the other requests that have been made or will be made.

The criticality of the request helps drive the schedule for designing and implementing the monitoring. It is also determined by the impact of the outage or failure this monitoring addresses and the likelihood of failure. Mission-critical systems and applications should be monitored more than and differently from the systems whose impact is low. If an event occurs every day, and it is related to a mission-critical system, that event should be prioritized at a higher level for monitoring development than an event that has never occurred even though it might be related to a significant event.

5.3.4 Identifying the target audience

Every event that will be monitored in the enterprise should have some known value and an agreed-upon course of action. All the parties concerned should be in agreement, from the operations person who views the event to the administrator who receives the problem management ticket and is expected to act on the event. The target audience for this event must fully accept and own the implications of this event. Otherwise, even building the monitoring for this event is a waste of time.

Even the so-called “best practices” events that can be implemented by a solution provider or the vendor of the monitoring solution will not be best practices in an organization if they fail to achieve recognition as such. The administrator in company A might 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 in company B might think otherwise, and promptly disable the monitor after it alerts him twice in one day. Both experiences and activities can 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, usually results in wasted resources or dissatisfied customers. 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, let us say that a monitor is created for monitoring the levels of computer system activity by examining the queue lengths, processor activity levels, and so on. The event is generated and results in a problem management ticket.

The administrator of that system must first agree that activity level is something the person wants to be alerted about and that it is a problem on which the person will act.

If the administrator who receives this event is not willing to action the event, the reception of the event is a nuisance. In environments where all the concerned parties do not agree on the events that should be considered important, there is no clear value in the monitoring solution.

Such an organization usually has many monitoring solutions, each owned by a different faction of the organization, and each believed to deliver the value the supporting faction seeks, regardless of the ability of a single tool to meet the needs of the entire organization.

5.3.5 Identifying and refining the possible events list

The first step in designing 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 are the business views, links, workspaces, and reports that are required, and for which communities?

Plan the management of events and the automation to solve the monitoring needs, keeping in mind the changes in the volume of events and the impact on the support and operations areas. Experts from each area should be brought in to help with this design.

For every event generated by every application and every system in the environment, some relationship with other events or occurrences in the environment is likely to exist. Consider a request to monitor this event:

```
EVENT_ID 20503The application WYSIWYG has failed with error -9.
```

This can be seen as a possible error to alert about 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, these can be understood only with experience. Even though picking up these alerts for the purpose of validation can be a worthwhile endeavor, the meaning of the event should be understood so that an appropriate correlation can be made and action taken to resolve the issue.

Just as it is important to identify the specific indicators 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 or require significant resources beyond a sustainable or desirable level.

The monitoring team usually works with the team supporting the enterprise tools to ensure that their desired monitoring solution is technically feasible and supportable. For example, an application developer who feels that event 20503 must be trapped to correlate might need some help in understanding the impact of doing so if these events are thrown at a rate of 100/second. This requires some changes to the architecture and possibly other components as well.

5.3.6 Meeting the target audience for approval

Before implementing the monitor in the enterprise environment, complete all the corresponding activities and obtain sign-offs on the following aspects:

- A business value is defined.
- Owners have been identified, and they agree to the ownership.
- An appropriate system management entity (SME) has been identified and consulted, and the person concurs that the request is a valid measurement.
- The possible action to be taken in response to the reception is decided on.
- The actor responsible for the action is identified and is aware of the event and agrees to action it when notified. The action can be automated, if possible.

At this point, the monitor does not exist. Before allocating resources to develop the solution, all these sign-offs must be in place. Failure to do so is likely to end in dissatisfaction and the inability to recognize or achieve the potential of the solution.

5.3.7 Creating, testing, and implementing the monitor

Segregate the duties first. The creator of the monitor is *not* the tester, the implementer is not the tester, and so on. All these people should be clear about their responsibilities in relation to this monitor.

A small amount of work is likely to remain when the solution becomes ready for production and for final user acceptance. In all probability, a certain amount of customization will take place to accommodate the final or late identification of needs. Do not forget to develop the necessary 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 the product functions, you can make changes in the production environment whenever you work to create monitors if you are not working on an isolated system. Doing so means inviting trouble, and we recommend *against* it.

Lock down security in the production environment to the point that only a very select group can make changes to the situations running in production. Any changes in the production environment should 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, 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 a previous point if you accidentally introduce a problem in the future.

The facility that enables you to export situations from your test environment to re-import later or to import into your production environment is a set of command-line commands displayed in Example 5-1.

Example 5-1 Command-line interface: Export situation

```
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 Extensible Markup Language (XML) file into the test environment later or into the production environment is done through the facility to re-import these XML files, which are essentially command-line commands, as shown in Example 5-2.

Example 5-2 Command-line interface: Import situation

```
tacmd createSit {-i|--import} [FILENAME]

-i|--import: Specifies the situation definition to import.
```

5.4 Building the monitor

While building the monitor within IBM Tivoli Monitoring Express V6.1, you have a choice between the Situation Editor and the command line. This section first discusses building monitors in the Situation Editor and later the graphical user interface (GUI). This is because when you look at the GUI, you will see the formula that you can build from the command line. It is easier to understand the options if we explore the Situation Editor first.

Building a complete situation involves the following considerations:

1. Naming the situation
2. Selecting attributes for comparison
3. Editing the formula
4. Selecting targets of distribution: MSL, systems, or both
5. Writing expert advice to explain the situation event
6. Setting the action
7. Using the Until tab
8. Best practices for situation creation

We followed these steps when creating one situation each for DB2 Universal Database (UDB) and Microsoft Windows Active Directory, so each topic appears for both components.

5.4.1 Naming the situation

Launch the Situation Editor from the Tivoli Enterprise Portal browser or the desktop by pressing Ctrl+E. This opens the Situation Editor window, shown in Figure 5-2, which provides a list of situations in your monitoring environment.

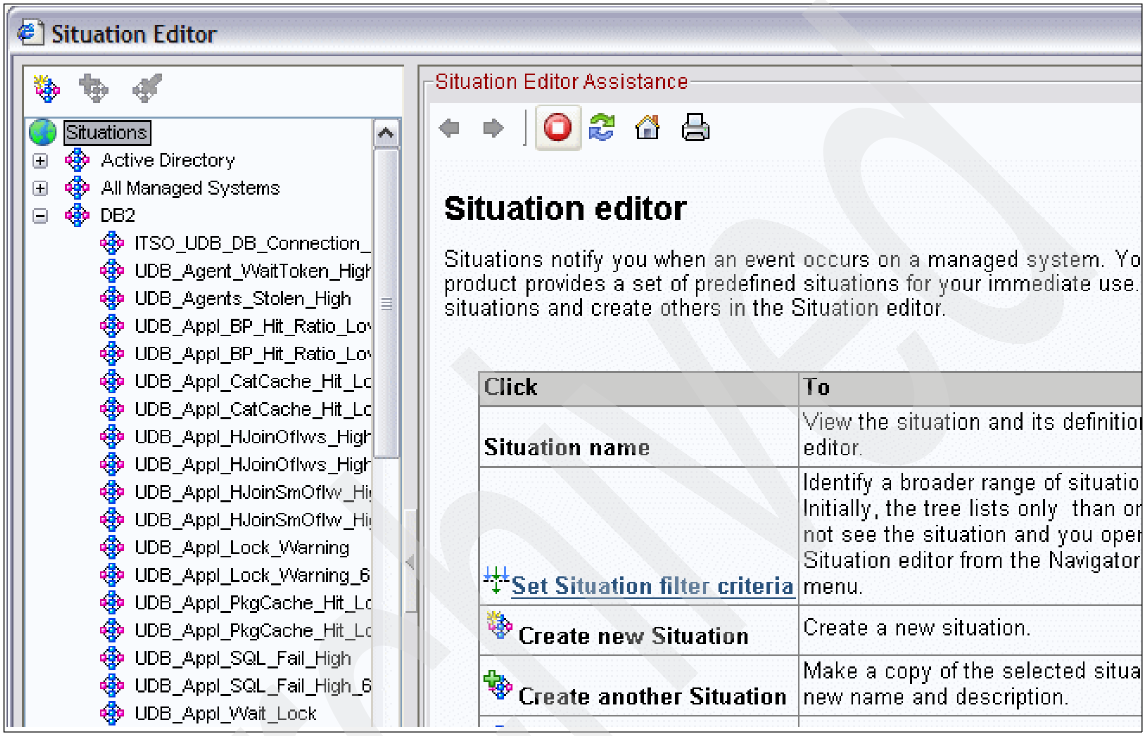


Figure 5-2 The Situation Editor

Take a few moments to explore the types of situations that exist out of the box. In our case, we looked for a situation that monitors the status of database connections.

Tip: Besides our choice of Ctrl+E, it is possible to launch Situation Editor in several ways from the Navigator.

If you right-click the enterprise item and select the **Situation Editor** icon, you will see the situations that are distributed to the monitoring server.

If you right-click a system item and select the **Situation Editor** icon, you will see all the situations that are distributed to that managed system.

If you right-click an agent item or if the agent has subagents, you will see all the situations that are distributed to the monitoring agent. At the attribute level, you will see situations distributed to that managed system, but only those that were created with related attribute groups.

With Ctrl+E, you will see the realm of options in the environment. This option helps build situations that are not associated with a Navigator icon.

Let us, for example, presume that you need an alerting mechanism to tell you that a database is inactive. A mission-critical application in your environment is found to be having severe problems, which results in this failure.

The first thing to do is name the situation. In this example, we clicked the **+** (plus) icon to expand the DB2 category. We then clicked the DB2 title for the category and right-clicked the icon above to create a new situation.

Figure 5-3 displays a Create Situation window.

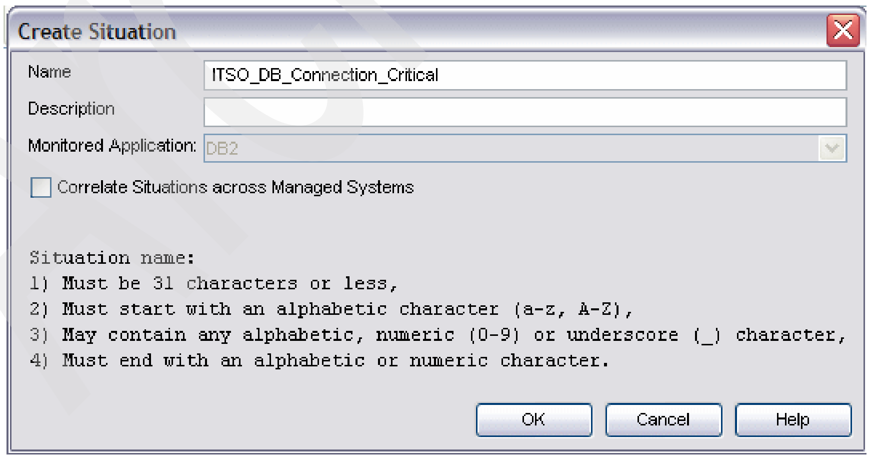


Figure 5-3 Creating the situation

When naming a situation, keep a few criteria in mind. First, look at the situation names that are provided as default with the product. Note that DB2 alerts begin with DB2_. However, it is useful to ensure that all your situations stand out by having your company's name at the beginning. An example format is as follows:

```
<Customer>_<Component>_<Description>_<Severity>
```

If your company's name is ITSO, and the component you are interested in is the database component, you can name it ITSO_DB_Connection_Critical. The Description field you populate in Figure 5-3 on page 214 populates the fly-over text of the situation in the Navigator.

You can also name the situation with a responsibility group. DBADMIN_DB_Conn_Critical is a good choice. The reason for doing this might be that your organization cannot standardize on naming the situations, even though this is highly recommended, because one size can fit all if you can make the involved parties agree.

Naming them with the same prefix makes them appear in the Situation Navigator in alphabetical order grouping as well. The format for this is:

```
<Responsibility>_<Component>_<Description>_<Severity>
```

Do not cross the 31-characters limit for situation names. You might, therefore, have to create abbreviations. For example, choose PWC_DB2 over PRICE_WATERHOUSE_COOPERS_DB2.

> **Note:** One product consideration is that if you save the name by closing the window shown in Figure 5-3, you will be forced to copy the situation to rename it. The function of duplicating a situation is called Create Another. It is essentially creating a duplicate copy. The first thing it does in the Situation Editor is provide you with a dialog box to name the copy (you can use this to rename, too). You will then have to delete the version with the name you wanted to change.
>
> This is a particularly good reason for developing your naming standards before you 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, and then delete all the incorrectly named copies. This involves a lot of work for someone later after a standard is created.

5.4.2 Selecting attributes for comparison

After naming the situation, select attributes for comparison in the situation. Figure 5-4 shows that the named situation has appeared in the list of situations.

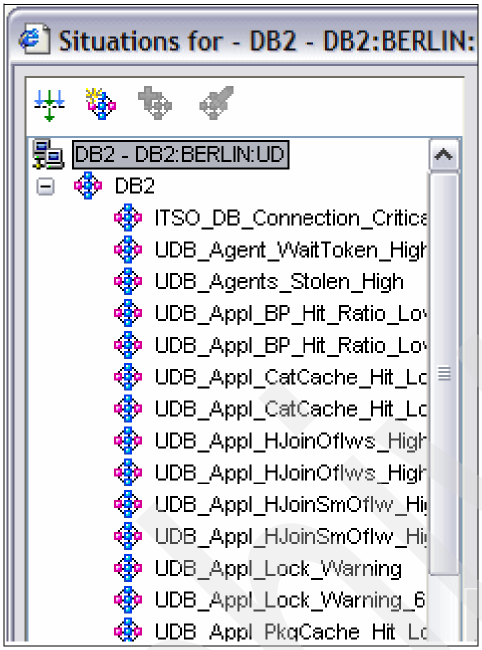


Figure 5-4 New situation name in the Situation Editor list

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.

A list of attribute groups are available for you to choose from. If you are not familiar with these attribute groups and what data each group represents, one approach is to view them from their respective workspace by performing the following tasks:

1. Expand the + sign under the DB2 agent name from the physical tree.
2. Select **Databases**. The workspace you see is the default workspace.
3. Right-click **Databases**. You will get a selection of workspaces.

4. Familiarize yourself with each workspace to get a better understanding of the attributes available to use in your situation.

   In our example, we created a situation that will alert us when a database is inactive. Therefore, we select the attribute group **KUDDBASEGROUP00** and the individual attribute **dbase status**, as shown in Figure 5-5.

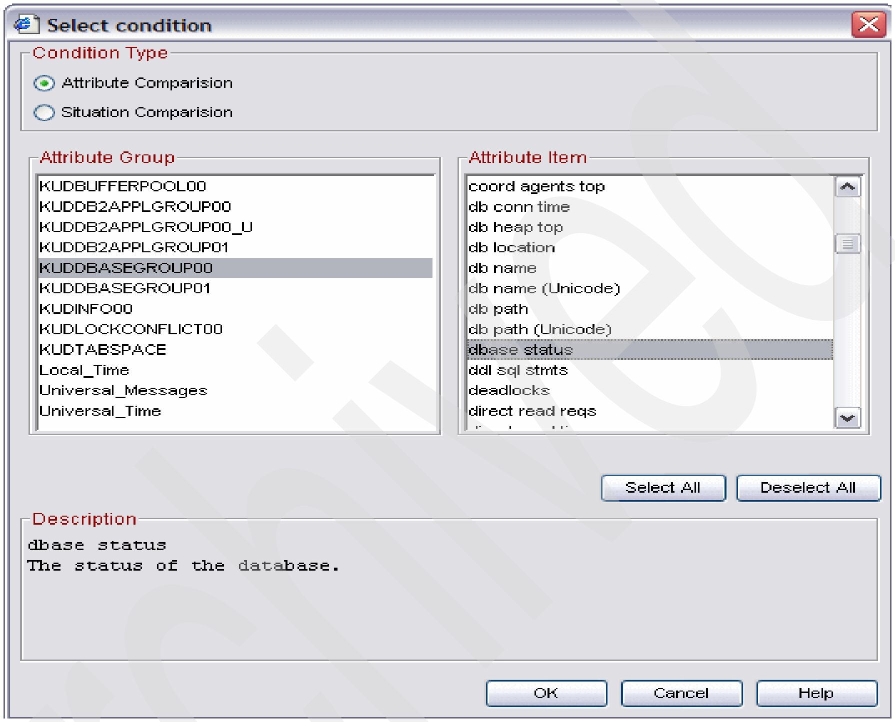


Figure 5-5 Select condition

5. When you click **OK**, the editing window opens.

> **Important:** There is a limit on the number of attributes you can select from this list that is based on the size of the formula that is created with the selection and creation of criteria for those attributes. Earlier, the number of attributes that you could have was a hard stop of 10 in the OMEGAMON product. That restriction has been lifted and replaced with a status bar indicating the current size of the formula against the total size possible. The current formula limit is 1020 characters.
>
> However, we recommend that if you have multiple situations using the same attribute group, limit the number of attributes to 10 for performance reasons.

5.4.3 Editing the formula

Figure 5-6 shows the formula editing window.

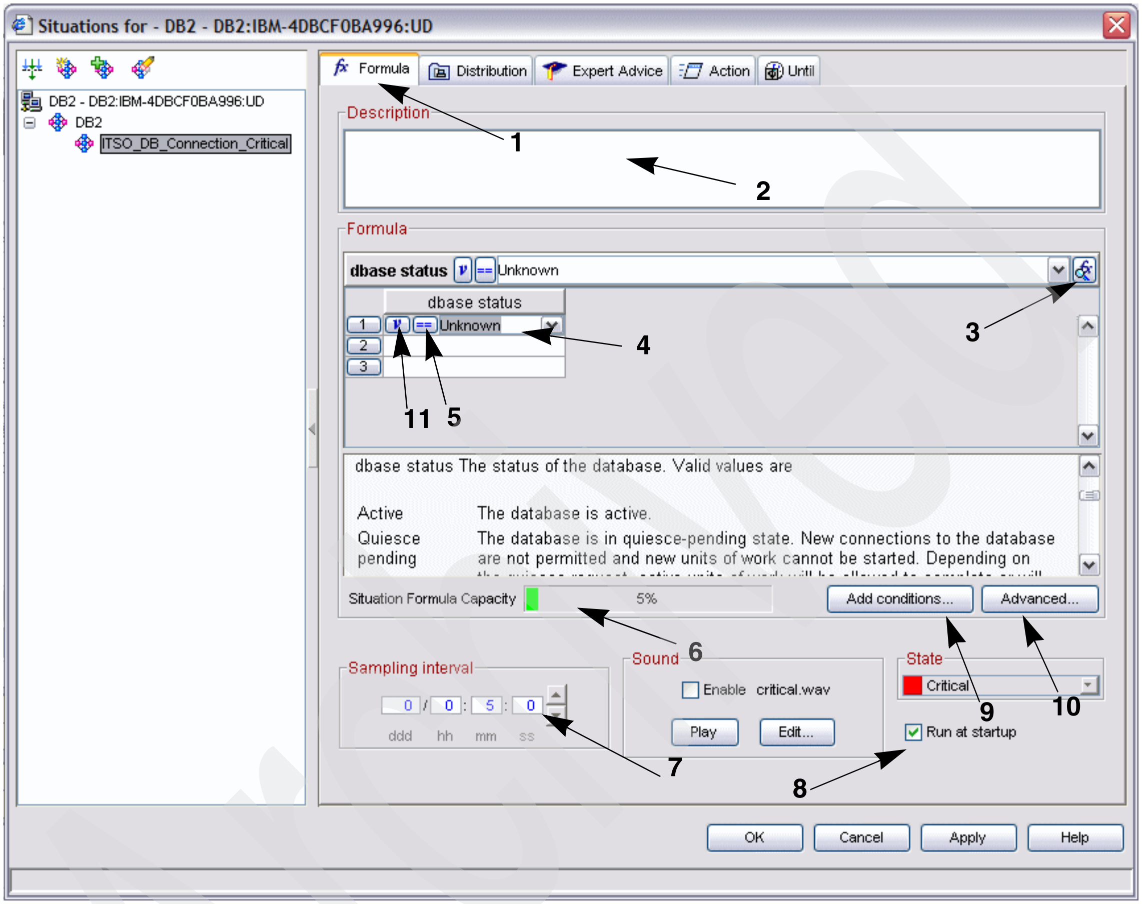


Figure 5-6 Formula editing

The following steps explain the steps in Figure 5-6:

1. Select the **Formula** tab.
2. Enter a short description here, which is limited to 31 characters.
3. Click this button to show the completed formula.
4. In this cell, type the value for the attribute or use the drop-down list to select the value.
5. Click this button to open the equality choices list (not equal, equal, grater than, and so on).

6. This shows the formula capacity. The total limit is 1020 characters. The bar shows the percentage used by you. (Refer to details provided in step 3.)
7. Set the sampling interval here. The default is 15 minutes, but in this example, we used 1 minute.
8. **Run at startup** is selected by default. If this option is selected, as soon as you save the situation, it starts running to all subscribed systems.
9. Click **Add conditions** to open the attribute selection window.
10. Click **Advanced** to activate two more options, Persistence and Display Item.

> **Important:** The Persistence setting requires that you wait for a selected number of occurrences of TRUE evaluations prior to creating an event. In IBM Tivoli Monitoring V5.1.1 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 will not alert you unless there are 10 in a row.
>
> The Display Item choice opens a window that lets you pick from certain attributes for which to continue checking in the sampling and create other alerts.

11. Click **Functions** for a list of available functions. This varies according to the type of attribute. The default is Value of expression. For a complete list of choices, refer to *IBM Tivoli Monitoring V6.1 User's Guide,* SC32-9409.

In our example, we used the status `= "Quiesched"`, set our sampling interval to five minutes, and set it to run at startup.

5.4.4 Selecting targets of distribution: MSL, systems, or both

Next we look at the Distribution tab for our options. Figure 5-7 shows the Distribution tab.

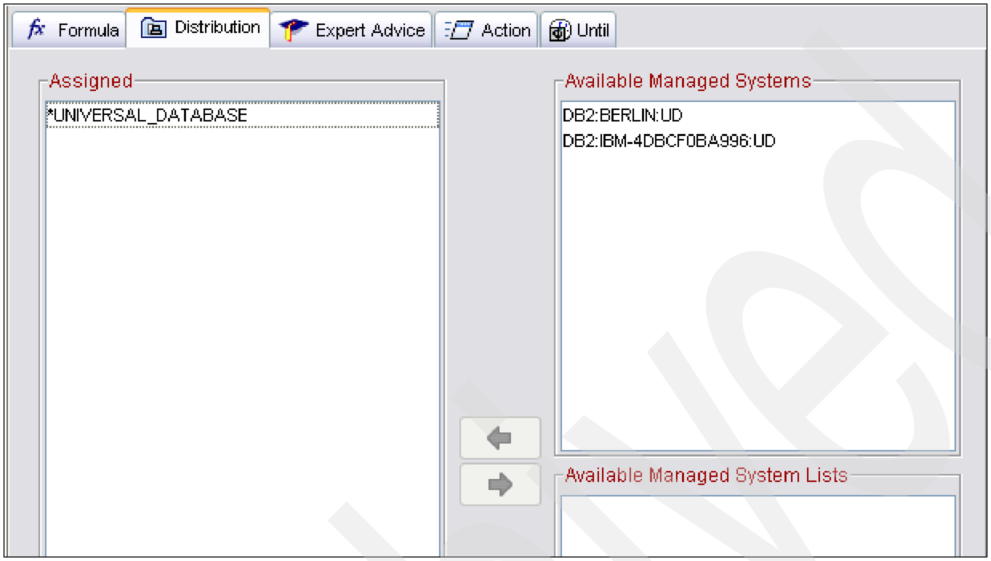


Figure 5-7 The Distribution tab

The Distribution tab offers the choice of single systems or managed systems lists (MSLs). These managed system lists are groups of servers and are defined by the Edit button. The MSLs that are predefined for you display an asterisk at the beginning of their name. Although the systems automatically go into platform managed systems lists by platform, you will have to move the systems into MSLs manually.

In this example, we picked the managed system list for *UNIVERSAL_DATABASE, so that all the new DB2 UDB systems added to the environment get this situation by default.

5.4.5 Writing expert advice to explain the situation event

The next tab in the Situation Editor is the Expert Advice tab, as shown in Figure 5-8. You can write text in the Expert Advice tab, and it displays this in the Expert Advice pane as text.

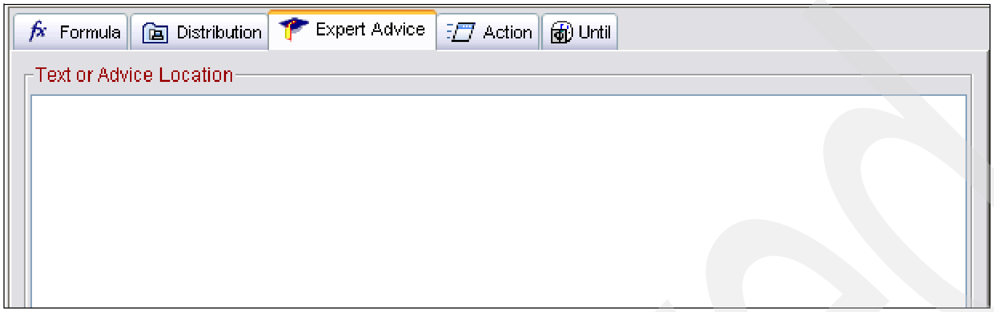


Figure 5-8 The Expert Advice tab

You can also write Hypertext Markup Language (HTML) code in this space. It will be handled appropriately. In this space, you can use the variable expression syntax such as the example shown in Example 5-3.

Example 5-3 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                  $EVENT:ATTRIBUTE.ISITSTSH.DELTASTAT$
```

To redirect the user to an existing Web page, simply include the URL in the pane without any other text, for example:

`http://www.ibm.com/developerworks/db2/newto/db2basics.html`

A browser opens and launches this page. You also can use variables if you want to launch the Web page in the context of a variable, for example, passing information to a search engine.

> **Important:** To avoid the 512-character limit, use an external Web file as the product does for the built-in situation expert advice. This does not, however, allow you to use the variable substitution. To use variable substitution, stay within the 512-character limit.

Perform the following tasks 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, as shown in Figure 5-9.

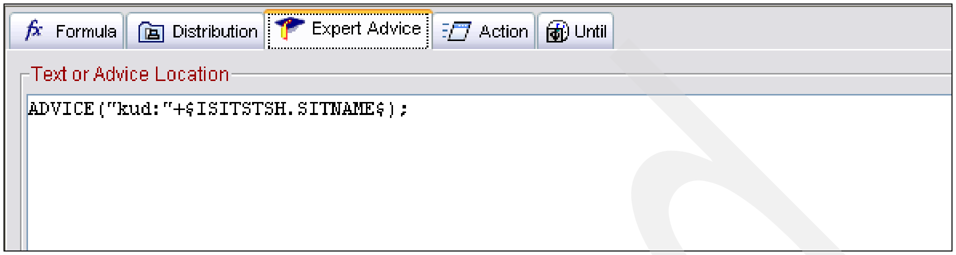


Figure 5-9 Populating the Expert Advice tab

2. Copy one of the IBM Tivoli Monitoring Express V6.1 expert advice HTML files, as shown in Figure 5-10, and create one with a name that matches your situation. In our case, we created ITSO_DB_Connection_Critical.htm. These files should be placed in the same directory as the other HTML files.

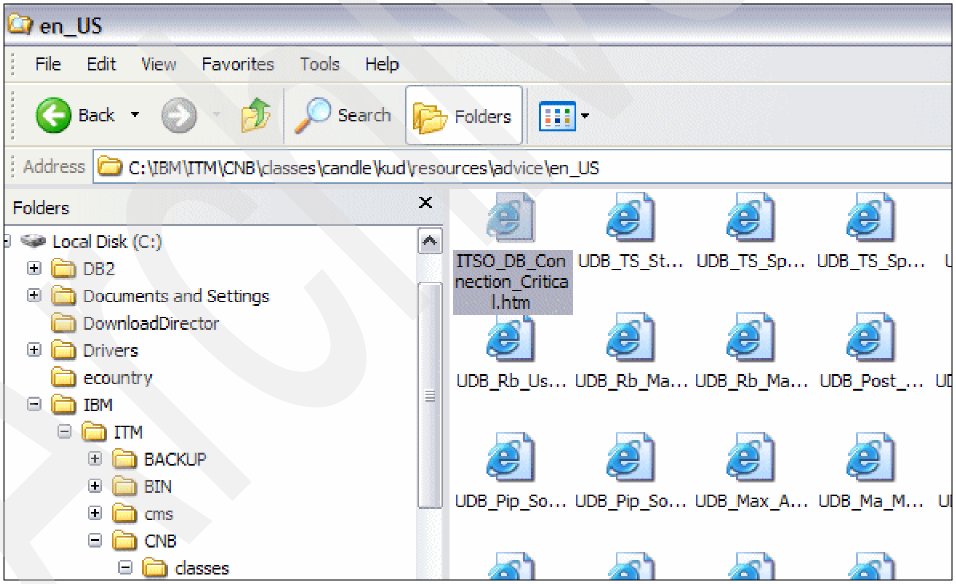


Figure 5-10 Default location of expert advice files on Windows enterprise portal server

3. Edit the expert advice file to contain the desired expert advice information. Save the HTML file.

4. Click **Preview** from the Expert Advice box to see how it looks (Figure 5-11).

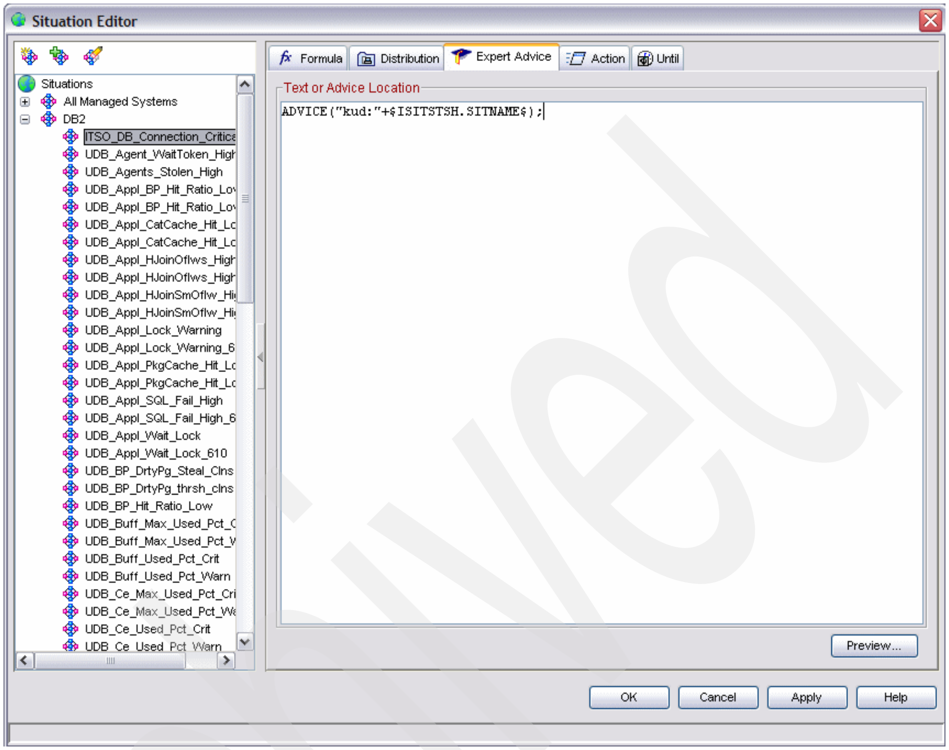


Figure 5-11 Preview of Expert Advice

A Preview window similar to that shown in Figure 5-12 opens.

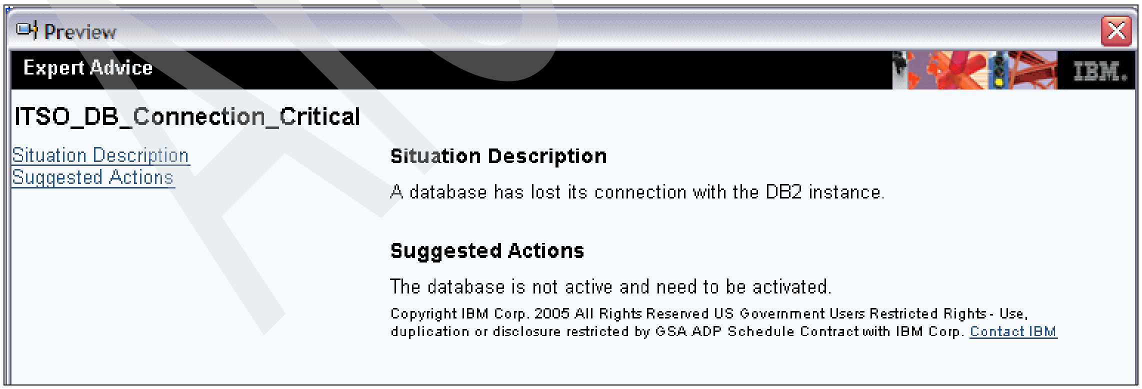


Figure 5-12 Custom Expert Advice for DB connection event

5.4.6 Setting the action

Use the Action tab to set the actions for when the situation becomes true, as shown in Figure 5-13.

Figure 5-13 Action tab with System Command selected

The Action tab offers two main options (Figure 5-13):

- System Command
- Universal Message

Figure 5-13 on page 224 shows the System Command choice. Figure 5-14 shows the changed section of the window if Universal Message is selected.

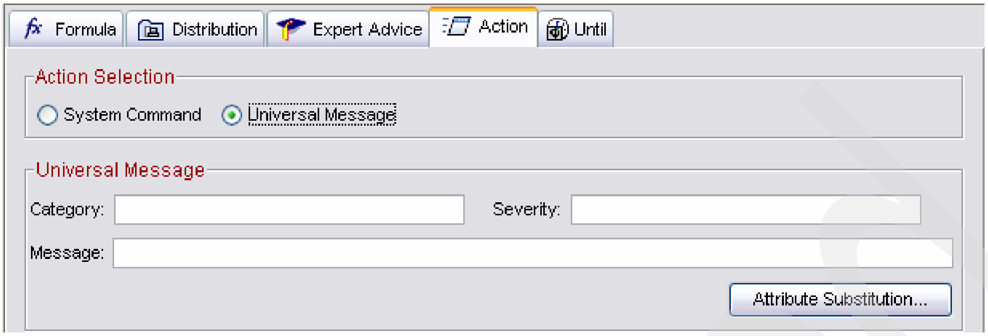


Figure 5-14 Action tab with Universal Message selected

Under the System Command selection, any system command can be issued. Clicking **Attribute Substitution** enters a variable so that when a situation is true, the appropriate values will be used:

```
echo " &KUDDBASEGROUP00.db_name on &KUDDBASEGROUP00.db_location is inactive
```

The system command should be platform-appropriate. An example of using the variable substitution is to keep the process name generic and thus have a more multipurpose situation.

Other options on the Action tab enable you to customize responses such as:

- Only take action on first item
- Take action on every item

Use the latter option when there are multiple matches to the same attribute condition check, thus returning a result of multiple rows of data. Selecting **Take action on each item** causes your system command to be executed for each line.

The next group of actions that you can carry out in the Action tab is selecting the place where the action is issued, whether on the agent machine or the monitoring server machine. This useful if, for example, you want to build a log file of these entries in a single file on the monitoring 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 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 issues the command with every sample that is true.

> **Important:** Pay attention to the sample interval. 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 because this will issue the command at least two times before the first attempt has had a chance to be successful.

Selecting **Universal Message** helps you define a universal message for the universal message log (UML). Define the category under which the message should fall, such as Non-essential or SiteCritical. Type a one-word term of up to 16 characters to categorize the message. Severity such as High or 1 is important to the message. Type a one-word term of up to eight characters that conveys the severity of the message. The 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 clicking **Attribute Substitution**.

5.4.7 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 do not want it to be closed automatically, do not choose anything on this tab.

The Until tab helps you perform two functions:

- 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 should not be manually closed. If they are closed, the situation goes into closed status. Only when this situation recovers or becomes false and then suffers again or becomes true will you see this event again. For example, if you set the NT_Missing_Process to look for Notepad on the system and set an Until for 30 seconds, with the Notepad not running in 30 seconds, the event closes and disappears soon after. Although the “critical” situation still exists technically, the tool cannot alert you until somebody starts the Notepad again and closes it.

5.4.8 Best practices for situation creation

Data for situations and events is collected at regular intervals. However, situations do not have to be active on a 24x7 basis often. For example, many alerts might only be required during normal business hours. The first way to control resource usage by situations is to stop them and start them only when they are required. To accomplish this, create 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.

Discuss with end-user departments about which situations should be built and how it is one of the most critical success factors of the project. If critical components are not being monitored by a situation or by not using the right thresholds, there is a risk of problems arising without being noticed.

If the situation intervals are set too short, the processor usage and network activity will be too high and the overall implementation might even become unreliable when the components cannot handle the workload any more, for example, if the Tivoli Enterprise Monitoring Server is evaluating more situations than it can handle or receiving more alerts, it will start to queue them, generating even more processor usage and network activity and delaying critical alerts from being raised.

If the situation interval is too long, problems might be detected too late, thus 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, the person might not be sufficiently aware of critical performance factors and might even lack sufficient authority to defend the outcome of the discussions with his department.

Discuss the following items:

- Critical performance factors for the application or system. These must be translated into data attributes to be monitored by using them in situations.
- The best values to check these attributes against. Should multiple situations be created to watch several levels of severity?
- If you 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. For more details, refer to “Situations grouping” on page 228.
- Select realistic alert values. For example, if a situation triggers and resets frequently, the Tivoli Enterprise Portal user in operations might get overloaded with alerts and lose reactivity over time. Moreover, this causes unnecessary processor usage and network activity to the Tivoli Enterprise Monitoring Server. Processing is required to handle the alerts and to store the data that leads to the alerts.
- The systems that should be monitored. Group systems into user-managed system lists. Situations will then be distributed to the managed system list. When a system has to be added for the same kind of alerting later, the only change required will be to the managed system list.

- The advice that can be given to the operator when the situation is triggered. 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 gets assistance in taking the right action that is consistent with the company's policies.
- The need for automated action, if any, and the action itself. This results in a simple command to be executed on the system (reflex automation in the situation).

Situations grouping

Grouping situations can save a lot of resources. Unfortunately, however, they cannot be set manually. The Tivoli Enterprise Monitoring Server decides whether to group situations. The following conditions must be met before a situation can be a part of a group:

- All situations in the group should use elements from the same attribute group.
- The situations must use the same interval setting.
- The situations must have autostart set to YES.
- The situations cannot contain an UNTIL clause.
- The distribution lists of the situations can be different.
- The situations cannot contain a display item.
- The situations cannot contain a Take Action item.
- The situations do not support the MISSING function.
- The situations do not support the SCAN and STR functions.
- The situations do not support event persistence.
- The situations do not support the group functions on the attribute criteria, such as average or total.

If a 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 the situations in the group will make use of the same data.

Situation grouping is done by Tivoli Enterprise Monitoring Server when it starts. If the monitoring server finds a number of situations that are eligible for grouping, it creates a new internal situation that performs data collection at the specified interval.

All the grouped situations then compare their criteria to the data returned by the internal situation. These internal situations only exist for the duration of the Tivoli Enterprise Monitoring Server run. They get an internal name that starts with _Z_ and the full name is built from the following parts: _Z_, table name, sequence number. For example, on Windows, when grouping situations on the table WTPROCESS, the grouped situation will be called _Z_WTPROCESS0. These situations are not added to the permanent situation tables such as TSITDESC in the monitoring server. However, because they are temporary, they can only be seen in situation temporary tables such as TSITSTSC.

To verify if any grouped situations have been created, run a SQL statement from a Tivoli Enterprise Portal view, using custom SQL:

1. Open the Query Editor by pressing Ctrl+Q (Figure 5-15).

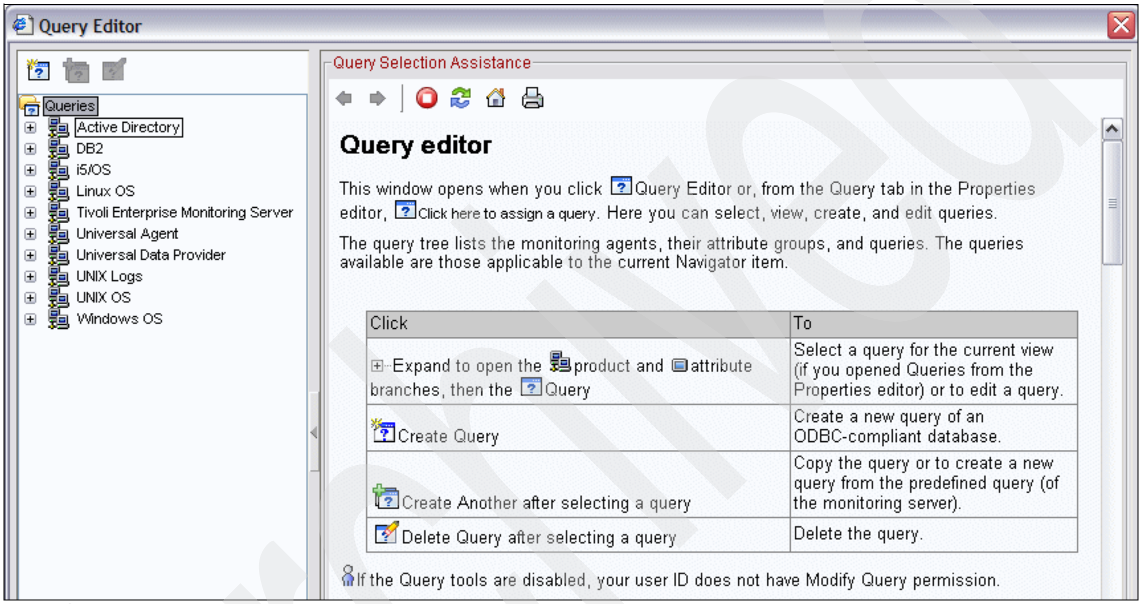


Figure 5-15 Query Editor

2. Click the **Create Query** icon.

3. In the Create Query window (Figure 5-16), enter the following details:

   – Name of the query.
   – Description. This is optional.
   – Select a category. We selected **Tivoli Enterprise Monitoring Server** in this example.
   – Select the data source.
   – Select the **Custom SQL** option.

   Click **OK**.

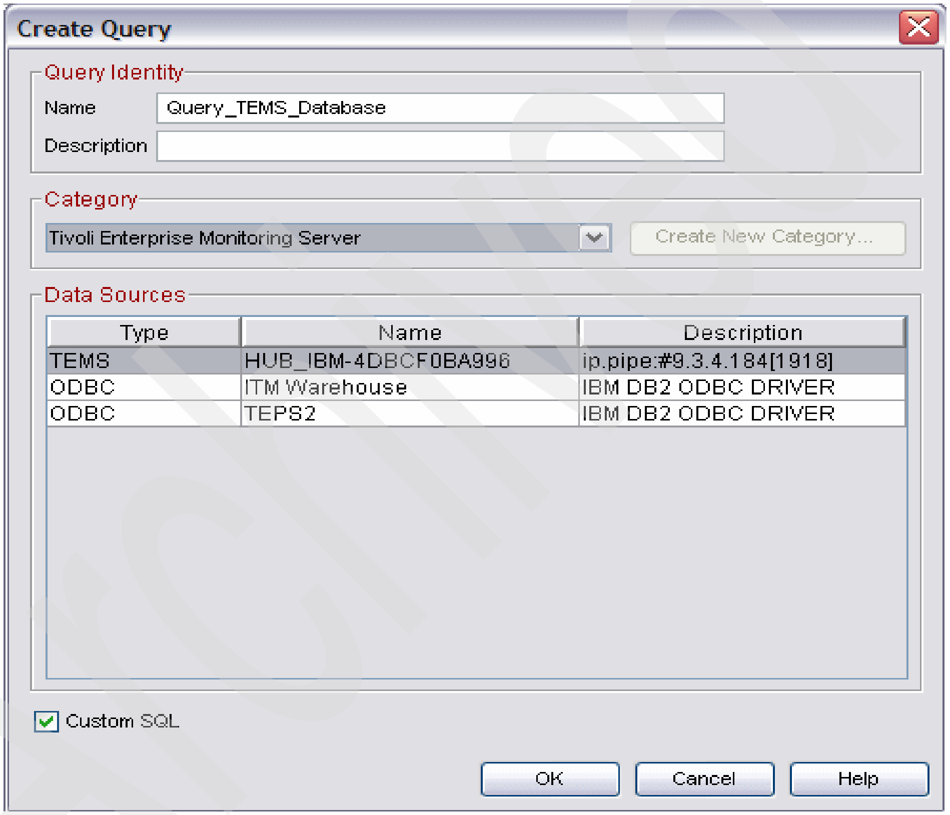


Figure 5-16 Create Query

4. Enter the SQL text in the next window (Figure 5-17) and click **OK**.

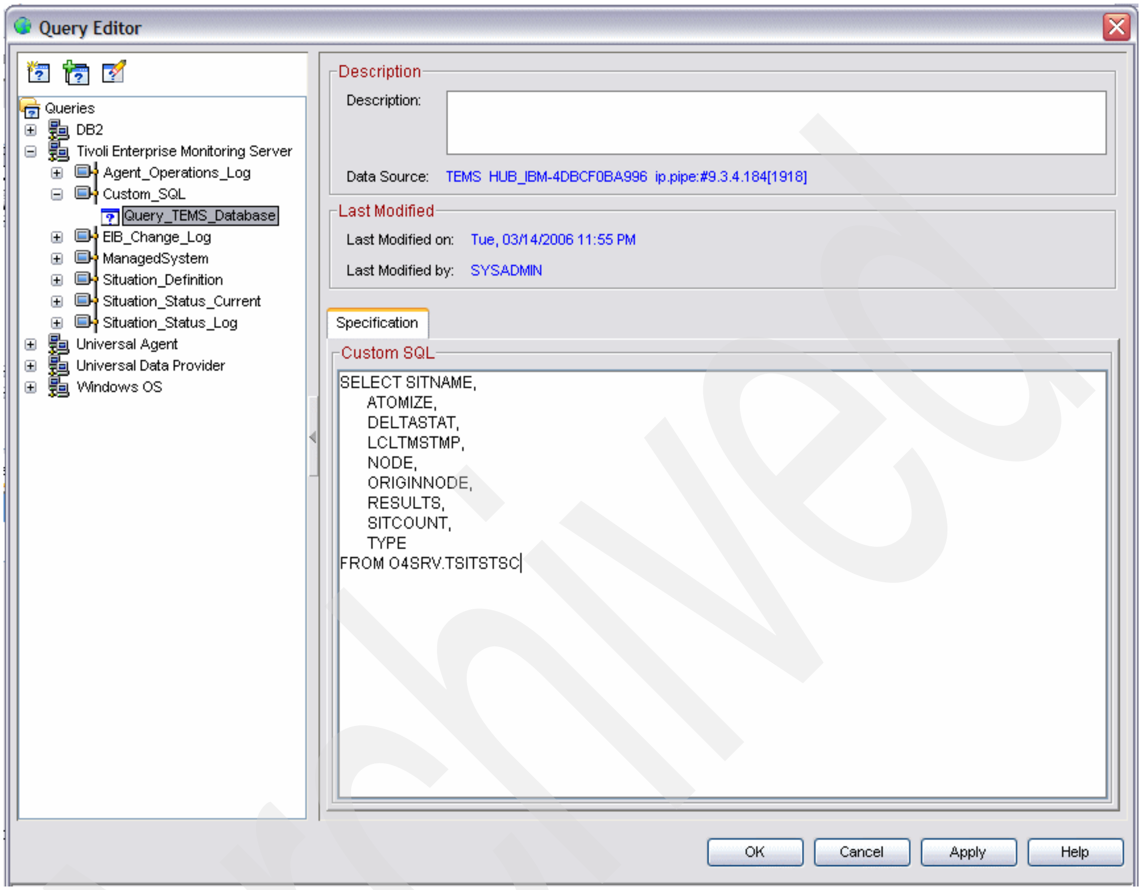


Figure 5-17 Entering SQL text in the Query Editor

5. When the query is used in a workspace, the result looks as shown Figure 5-18.

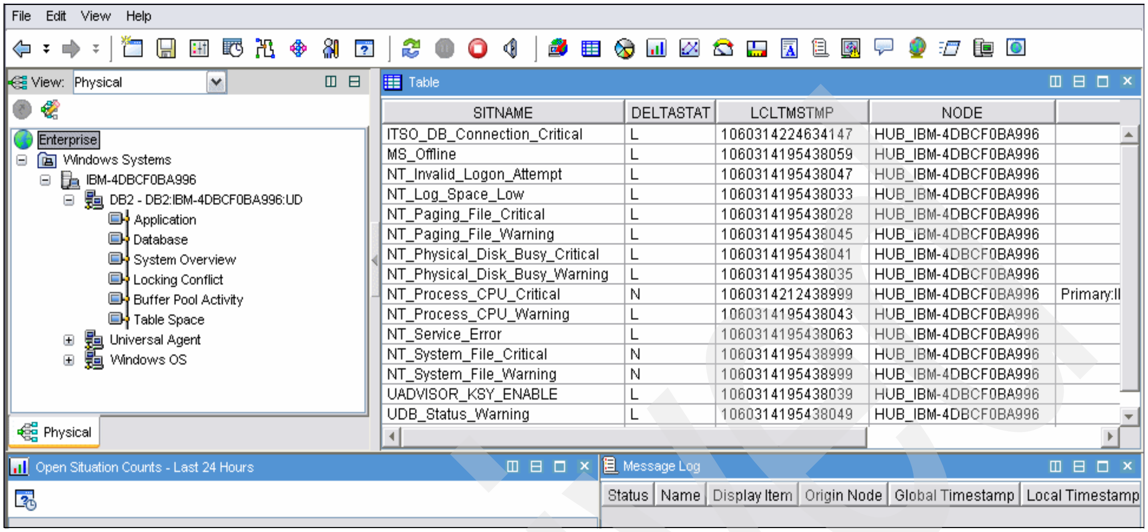


Figure 5-18 Using the query in a workspace

The grouping occurs only at Tivoli Enterprise Monitoring Server start-up, so any new situations or modifications will not benefit from grouping until the monitoring server is restarted.

Evaluating the situations

Situations can be evaluated at either the Tivoli Enterprise Monitoring Agent or Tivoli Enterprise Monitoring Server. Ideally, all situations will evaluate at the monitoring agent, as close to the data source as possible. Unfortunately, the monitoring agent is limited in its capacities to evaluate the situation. The evaluation will be moved to the monitoring server under the following conditions:

- If the situation has attributes that cross monitoring agents.
- If advanced checking is used, such as string scan.

If situations cannot be evaluated at the monitoring agent, the monitoring server will take over. Avoid evaluating situations at the hub monitoring server; all monitoring agents should report to a remote monitoring server.

Building a situation in the right order

When starting to build a new situation, make an overview of the attributes to test. Attributes will be tested from first to last, or from left to right on the Tivoli Enterprise Portal panel, in the order in which they are entered in the situation.

We recommend that you know the data behind the attributes. The first test to make should return as few rows as possible. The next step can further filter a limited set of rows. For example, on Windows, to check whether process XYZ uses more than n amount of real storage, test two attributes, that is, process name and real storage usage.

If you first test real storage use, the result set might contain multiple rows. Check whether your process name is among the returned rows. It is more efficient to first test on the process name, the result for which will be one row, followed by the test on the storage usage, on this single row.

Using complex conditions, such as string scan, sum, or average, can best be performed on a limited result set. Evaluate the attributes against simple conditions to reduce the result set.

5.4.9 Release management

After defining the situation and testing it to your satisfaction, put it into your versioning control system. Use the **tacmd** command line utility to export the situation.

> **Note:** If you issue the command without the **-e** parameter and simply redirect to a file, you are not exporting the XML. You are exporting a summary text to a file. Use the **-e** option to obtain the correct output.

After authenticating, issue the **tacmd** command to export the file. It is helpful to give it a version number. 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 made to the situation are later found to be undesirable and the previous state of the situation is more desirable.

Example 5-4 Exporting a situation to an XML file

```
[root@berlin][/opt/IBM/ITM]->tacmd login -s berlin -u sysadmin

 Password?

Validating user...

KUIC00007I: User sysadmin logged into server on https://berlin:3661.
[root@berlin][/opt/IBM/ITM]->tacmd viewsit -s ITSO_DB_Connection_Critical-e
/tmp/ITSO_DB_Connection_Critical_v1_0.xml

KUICVS004I: The Situation ITSO_DB_Connection_Critical was exported to
/tmp/ITSO_DB_Connection_Critical_v1_0.xml.
```

5.5 Building a monitoring view

Monitoring situations are only good and valuable when assigned to and viewed within a monitoring view (workspace). Although IBM Tivoli Monitoring Express V6.1 provides out-of-the-box predefined workspaces, it is always useful to create your own customized views.

Our goal is to build a business systems view (dashboard) of the Tivoli environment that monitors your infrastructure. You can have many different things in this view, but for purposes of example, we build a view that includes DB2 UDB and Windows Active Directory data, and uses existing and new situations and queries.

The steps to build a dashboard view typically involve the following steps:

1. Building a hierarchy of Navigator views
2. Assigning systems to the Navigator views
3. Building workspaces for views

5.5.1 Building a hierarchy of Navigator views

IBM Tivoli Monitoring Express V6.1 gives you an entire range of monitoring views out-of-the-box. These views gives you a good start in terms of monitoring your infrastructure because they provide important information about the components you have decided to monitor.

What is lacking is a more consolidated view of the infrastructure, including information about critical components that need to respond for the applications to respond to a user request.

One of the unique features in IBM Tivoli Monitoring Express V6.1 is the ability to create views that meet your personal or individual criteria. For example, a DBA might want to include information about all the databases connected within an instance, as well as information about memory and page file usage from the operating system where the databases reside.

Creating such views do not require any additional skills other than those you acquire upon completion of Tivoli training, more specifically about how to use the Tivoli Enterprise Portal user interface.

Figure 5-19 shows a consolidated view. This view provides information about how various components running on the same server are performing. We created a graphical view that shows the status on our two servers, Austin and Manchester. Each server has a Microsoft Windows OS, Windows Active Directory, and DB2 UDB agents running.

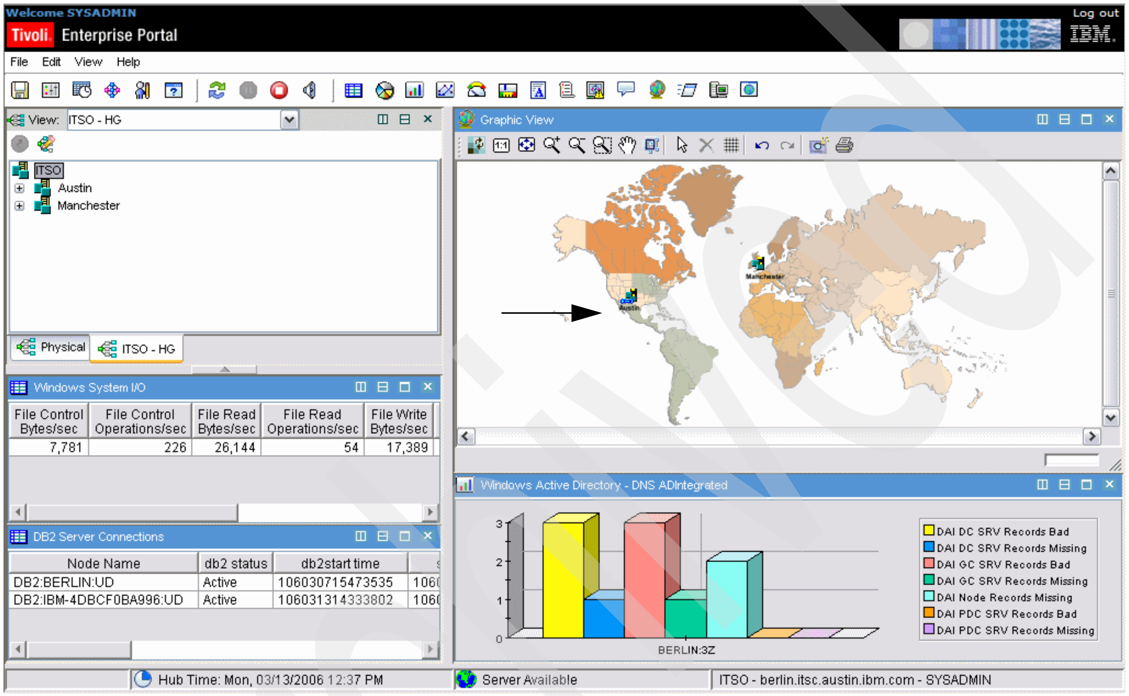


Figure 5-19 Consolidated Navigator view

The icon for Austin also has a blue paper clip, as indicated with an arrow in Figure 5-19 on page 235, that enables you to drill down to that server for further information about Windows OS and DB2 UDB performance on the Austin server, as shown in Figure 5-20.

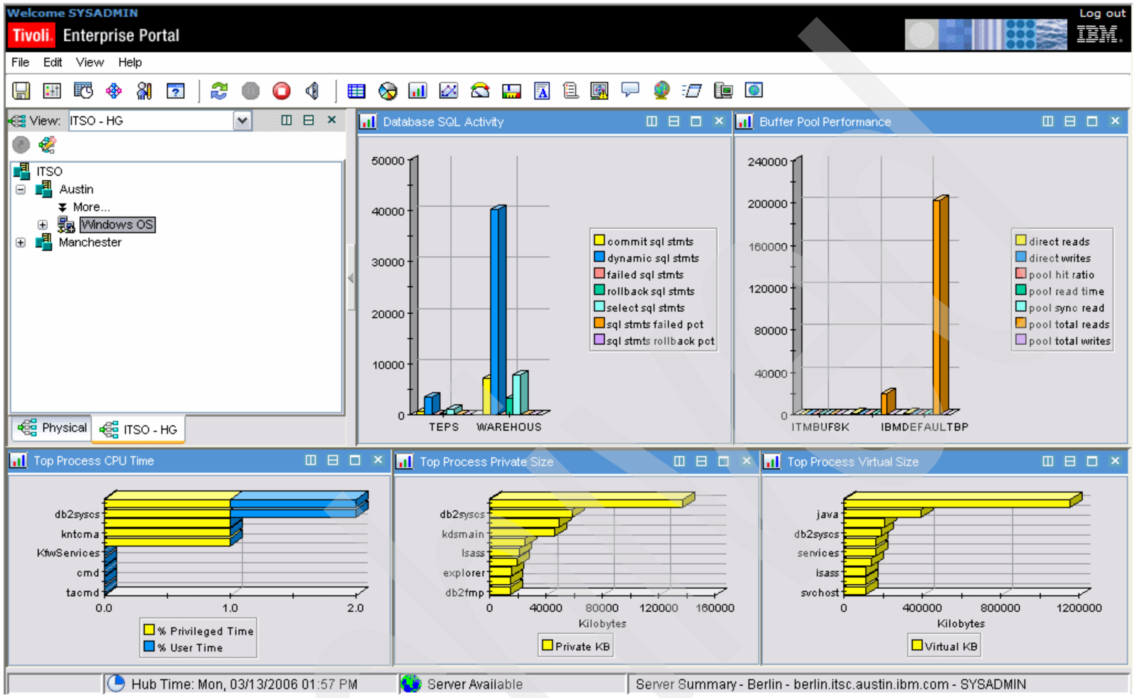


Figure 5-20 Server summary

Figure 5-21 gives an instant view of the following items:

- Top process CPU time, private and virtual storage size
- Database SQL activity
- Buffer pool performance

If we want more information about DB2 application performance, we simply expand the plus sign on the left side next to Austin (1) and select the DB2 agent (2).

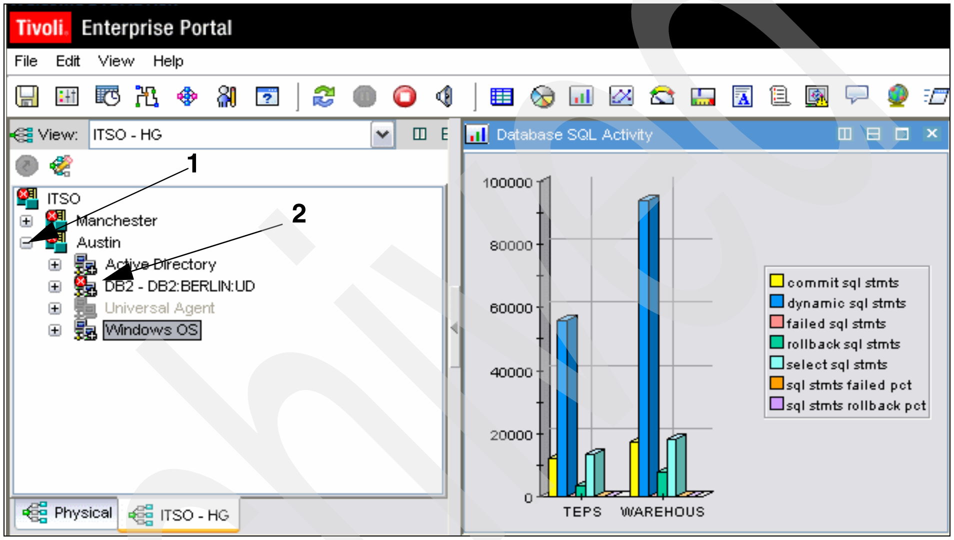


Figure 5-21 Server summary: Austin

Creating a Navigator view

Creating a Navigator view is not complicated and takes only a few minutes if you perform the following steps:

1. Open the Edit Navigator View panel to create a new Navigator, as shown in Figure 5-22.

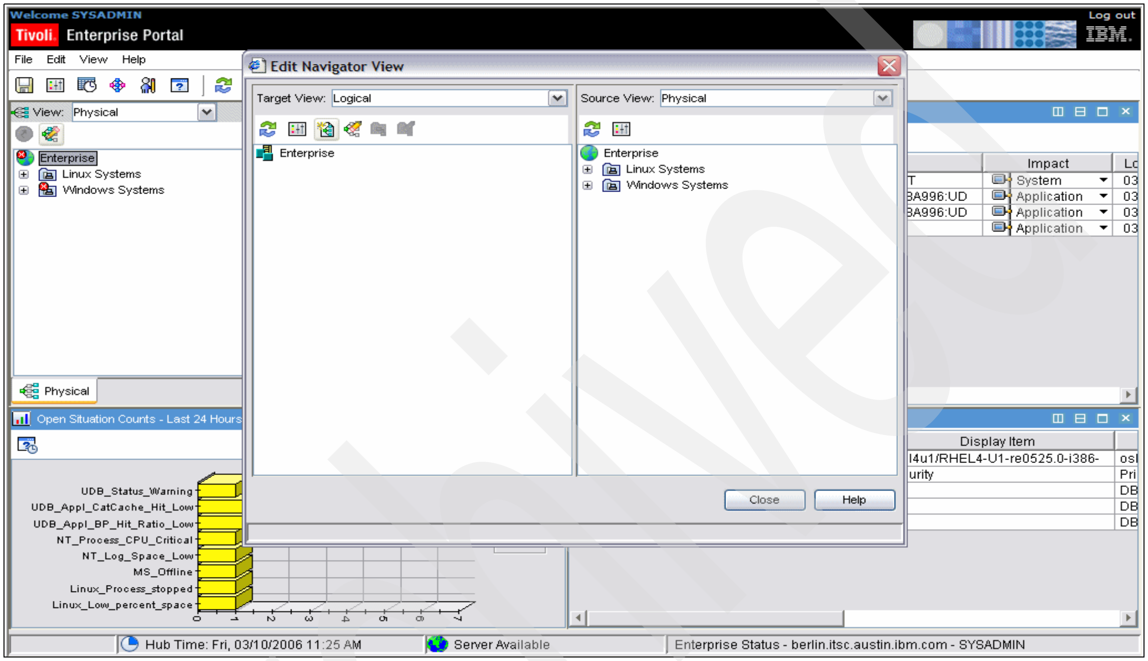


Figure 5-22 Open the Edit Navigator view

Note that the Logical Navigator is on the left. This is the target Navigator and can be modified. The Physical Navigator on the right is the source of all shareable items. It contains the Navigator items for all agents that have connected successfully to the hub Tivoli Enterprise Monitoring Server.

2. Select the third button from the left, as indicated by an arrow in Figure 5-23, to create a new Navigator view.

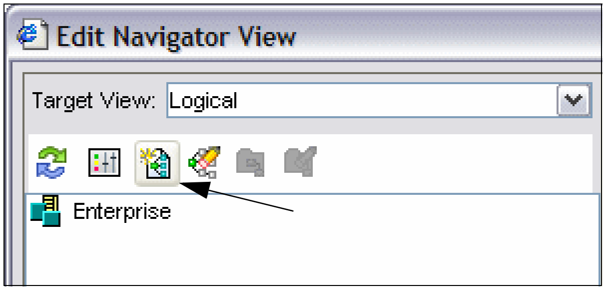


Figure 5-23 Selecting to create a new Navigator View

3. In this example, we added the Navigator name `ITSO Application Platforms` and gave it a description.
4. Create the structure of the Navigator. In this example, we built child items for the ITSO Navigator view to cover those areas. When you select the ITSO Navigator item, the Create child button, indicated by an arrow in Figure 5-24, becomes available.

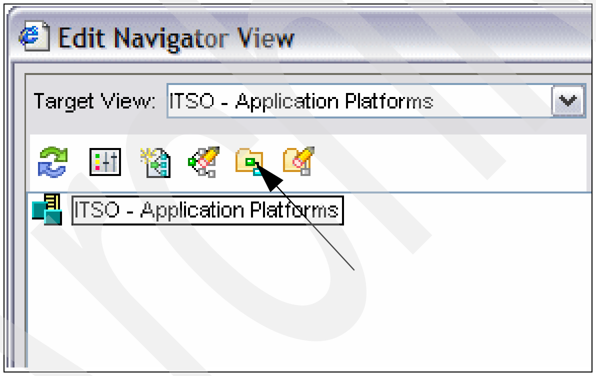


Figure 5-24 Creating a child item

Construct a hierarchy of Navigator views. In this example, we did not create all the workplace views for all the Navigator items in this book.

5. Create a child item that represents a physical location or name of a server. In this example, we used the name of the server in two different locations, Austin and Manchester (Figure 5-25).

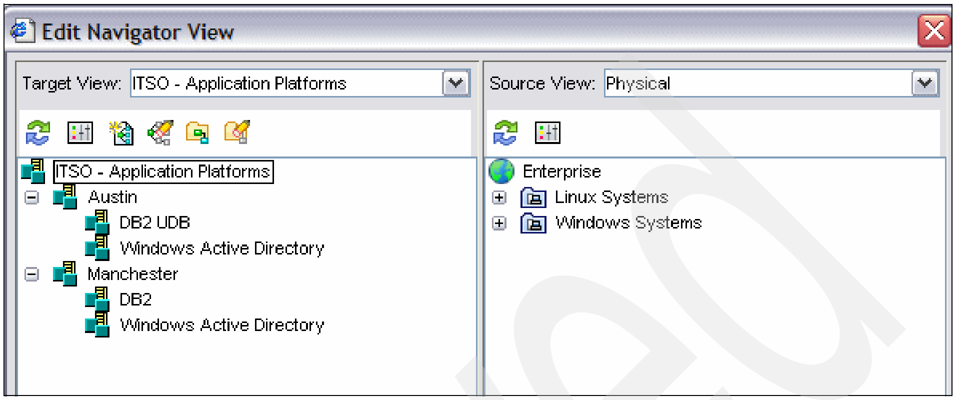


Figure 5-25 The Navigator View: Logical design areas

6. After building this structure, close the Navigator Editor. The work is automatically saved. You can now select this Navigator from the Navigator pull-down menu.

5.5.2 Assigning systems to the Navigator views

Assign the systems to each of these Navigator views. In this example, we started by assigning DB2 UDB in the infrastructure to the DB2 UDB Navigator by performing the following tasks:

1. Right-click **DB2 UDB** and select **Properties**.

2. Move the DB2 UDB into the Assigned area, as shown in Figure 5-26.

   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.

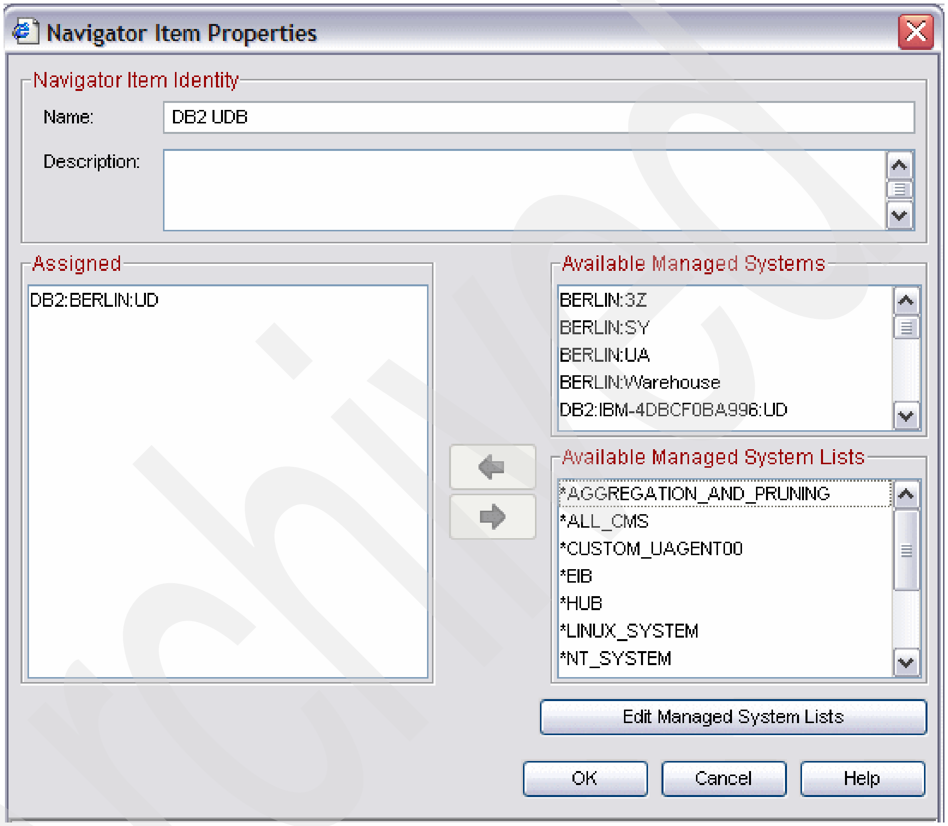


Figure 5-26 Moving a DB2 UDB server to the Assigned systems list

3. Repeat this process to add the systems to each Navigator area. Add the agent from a managed system that will provide the data you consume in that space.

After completing this process, click the Navigator items and start building the workspace view for these Navigator items.

If the child item represents an application that spans across multiple databases or operating systems, use the more generic *UNIVERSAL_DATABASE name for the managed system, rather than the unique name *DB2:BERLIN:UD.

5.5.3 Building workspaces for views

Before starting the construction of the workspace views, plan your workspace on paper. When you are building it in the Tivoli Enterprise Portal, it will be difficult and time-consuming 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.

In this section, we build the workspace displayed in Figure 5-27 by performing the following steps:

1. Click **Austin**. The workspace view changes to an unassigned default state (Figure 5-27).

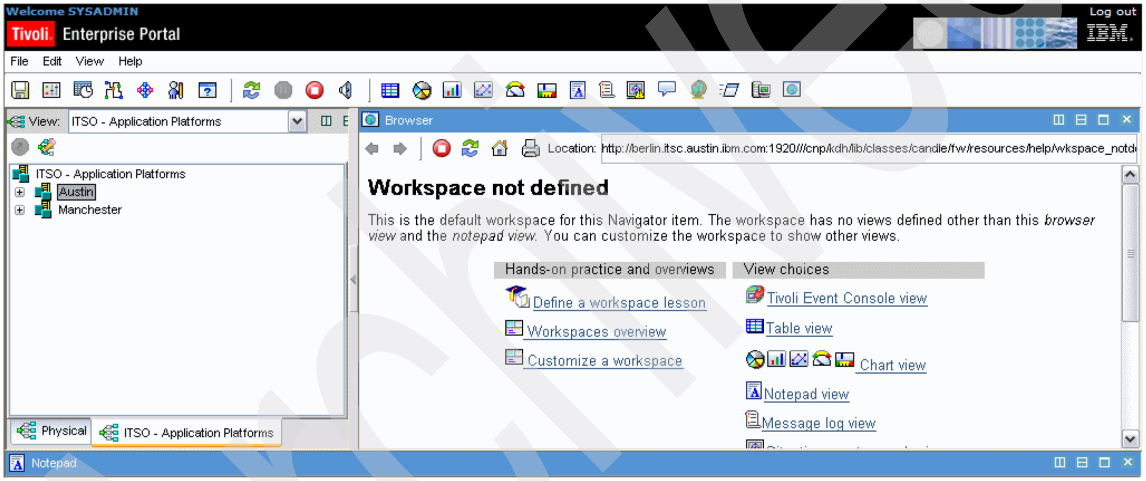


Figure 5-27 Default workspace view for our Austin server Navigator

2. Subdivide the real estate into the desired sections through vertical and horizontal splitting. Figure 5-28 shows our completed design.

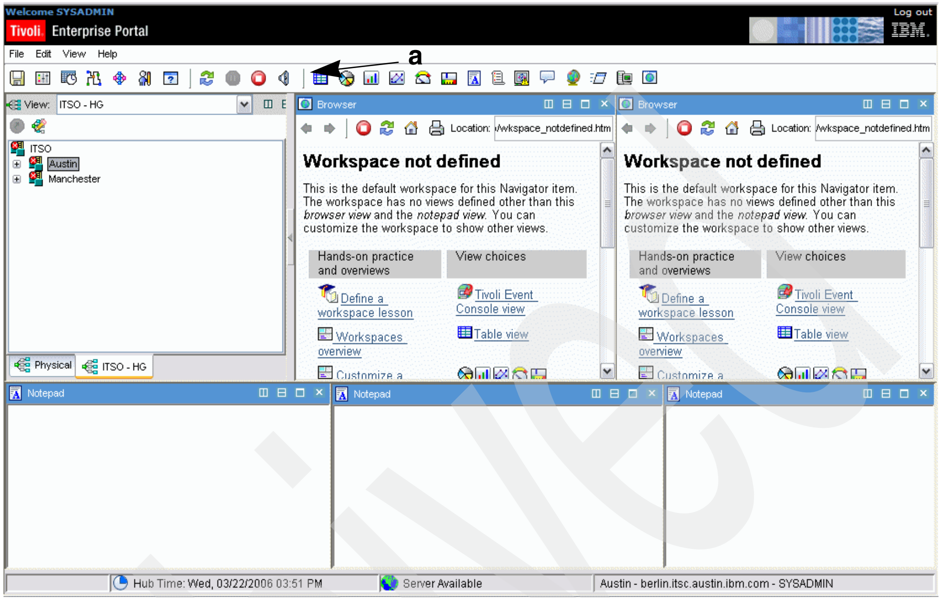


Figure 5-28 Default workspace subdivided

3. Next, assign queries to each pane. Click the Table icon (see the arrow in Figure 5-28), and then click the top-right pane. The Navigator view should now look like the view in Figure 5-29.

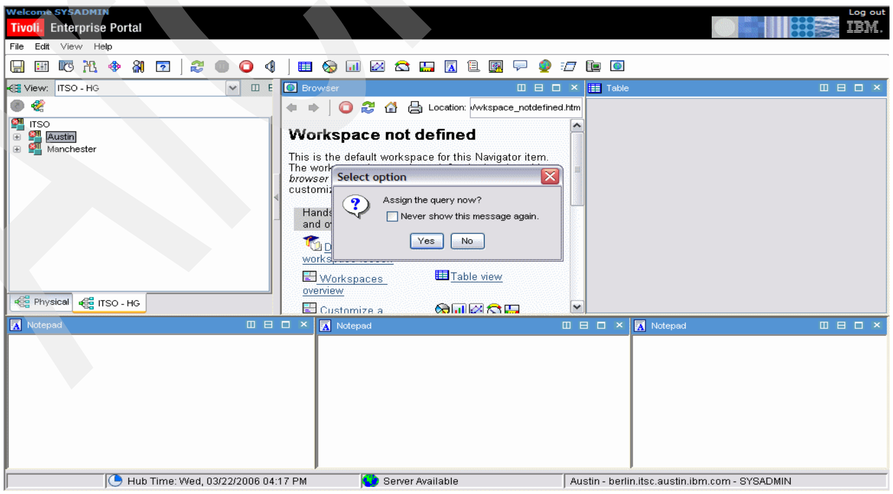


Figure 5-29 Default workspace: Assign queries

4. Click **OK**.
5. On the next panel (Figure 5-30), click **Click here to assign a query** (highlighted by the arrow in the figure).

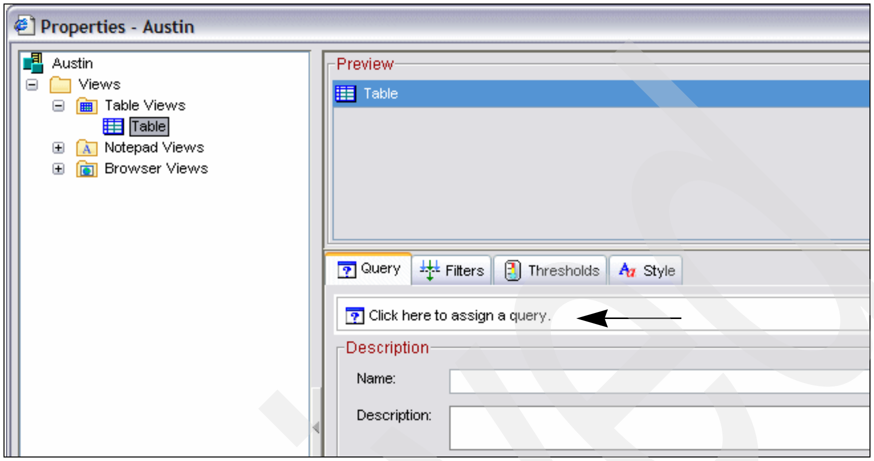


Figure 5-30 Default workspace: Select query

6. On the next panel (Figure 5-31), expand the DB2 icon to get a list of default queries.

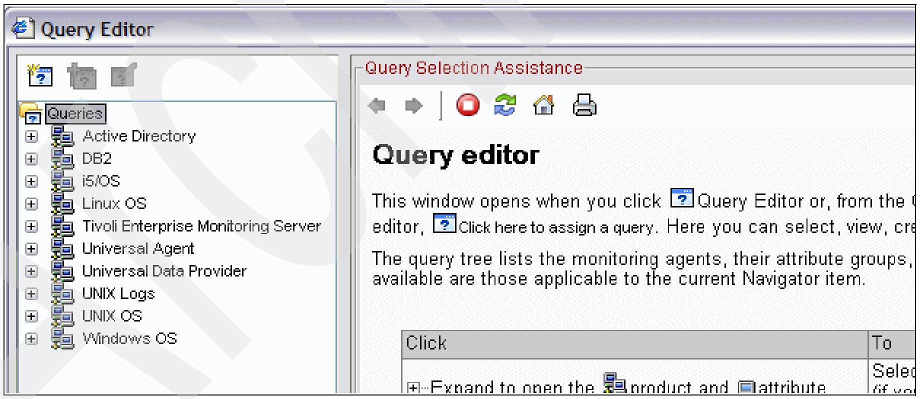


Figure 5-31 Default workspace: DB2

By expanding the DB2 plus sign, you will see a list of queries that we can use for our workspace. If you need to make any changes to these queries, you must create a copy of the query you are working with because it is not editable.

7. On the expanded list, select **KUDBUFFERPOOL00** and click **Buffer Pool Detail (Unicode)** (Figure 5-32).

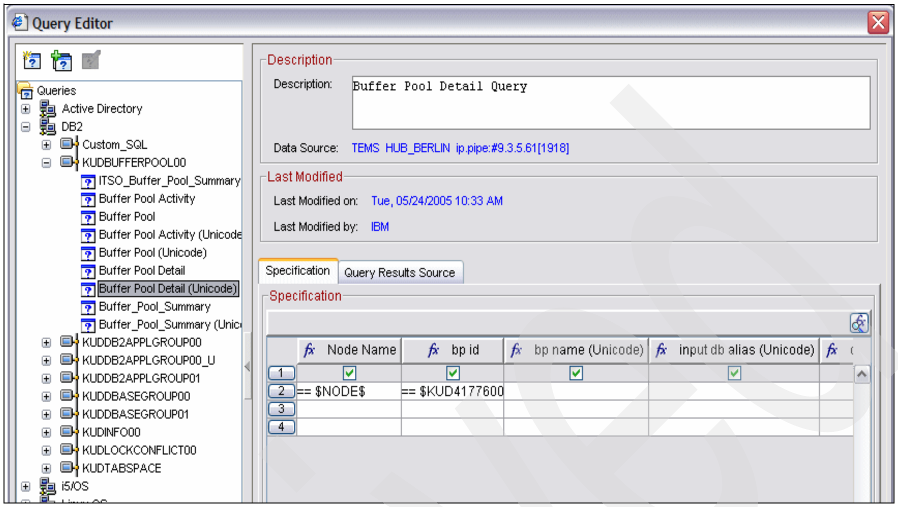


Figure 5-32 Default workspace: Buffer Pool Detail query

8. Because we already assigned the managed systems to the Navigator view, the correct agent should have been assigned to this query. To verify this, click **Query Results Source** (Figure 5-33).

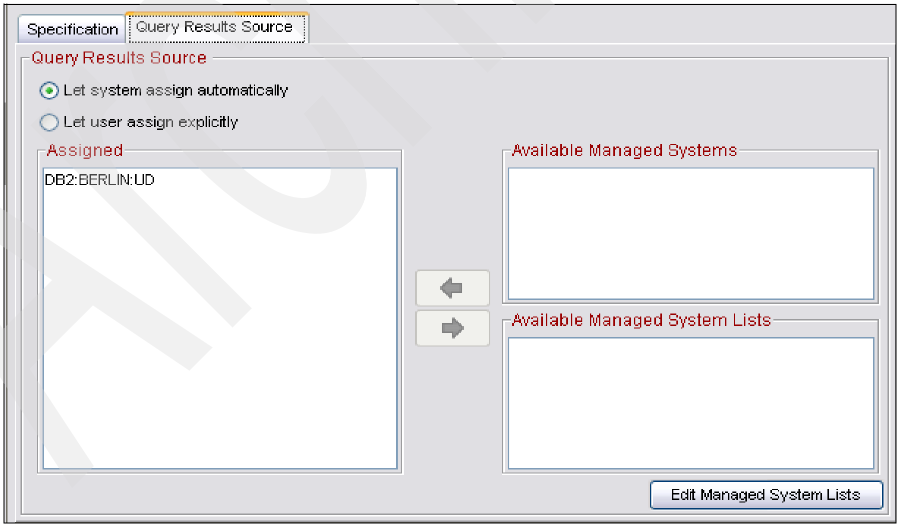


Figure 5-33 Default workspace: Query Results Source

As we can see, the system assigns the correct managed system automatically, and as we can see, it is the agent we want.

9. Click **OK** twice and to get the result as shown on the next view (Figure 5-34).

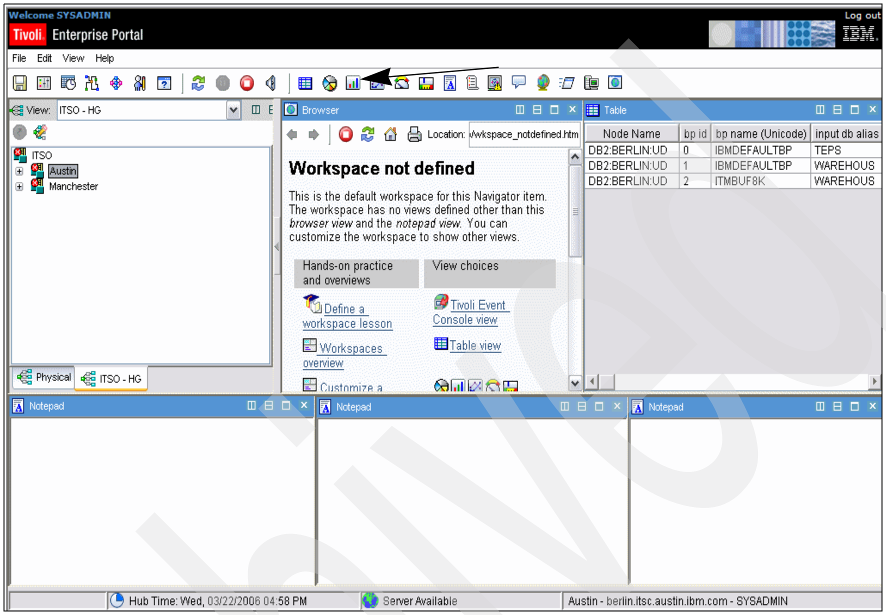


Figure 5-34 Default workspace: Table result

10. To change the table to a bar chart view, click the Bar Chart icon (highlighted with an arrow in Figure 5-34), and then click the table display.

11. On the next panel, we select the attributes we want to use for our bar chart. Because we will use more than one attribute, we use a combination of key strokes. Press the Ctrl key and click the first attribute direct reads, keep holding the Ctrl key and scroll to the next one, direct writes, click and repeat until you have selected all the attributes for our bar chart. The ones we use are direct reads, direct writes, pool hit ratio, pool read time, pool sync read, pool total reads, and pool total writes (Figure 5-35 on page 247).

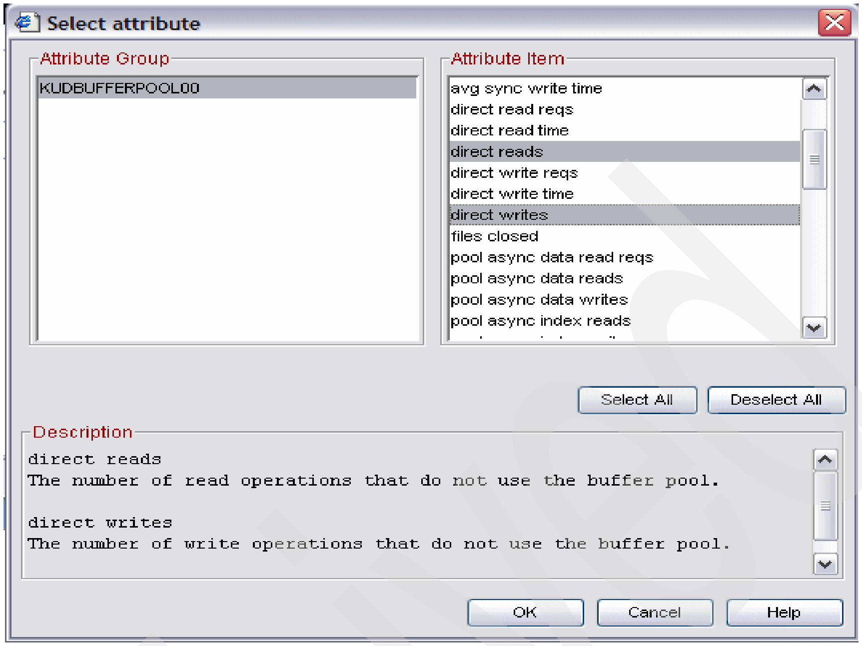


Figure 5-35 Default workspace: Select attributes for bar chart view

12.Click **OK**, and you should see a different view (Figure 5-36).

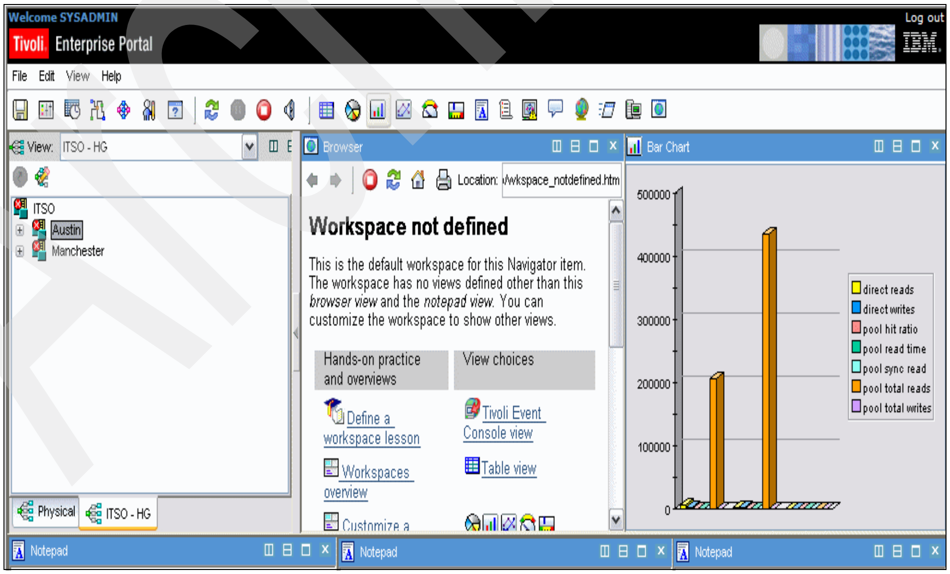


Figure 5-36 Default workspace: Bar chart view

We are almost done. This view is missing the name of the objects (buffer pools) and a title for the chart.

13. Right-click the bar chart and select **Properties**.

14. On the next panel, click **Style**.

15. On this panel (Figure 5-37), there are two things we want to do:

    a. Change the header. Click the header and enter the name of the chart to `Buffer Pool Performance`.

    b. Change the category axis. Click the category axis and then click the **Category Axis** tab again and you should now see the same view as in Figure 5-38 on page 249.

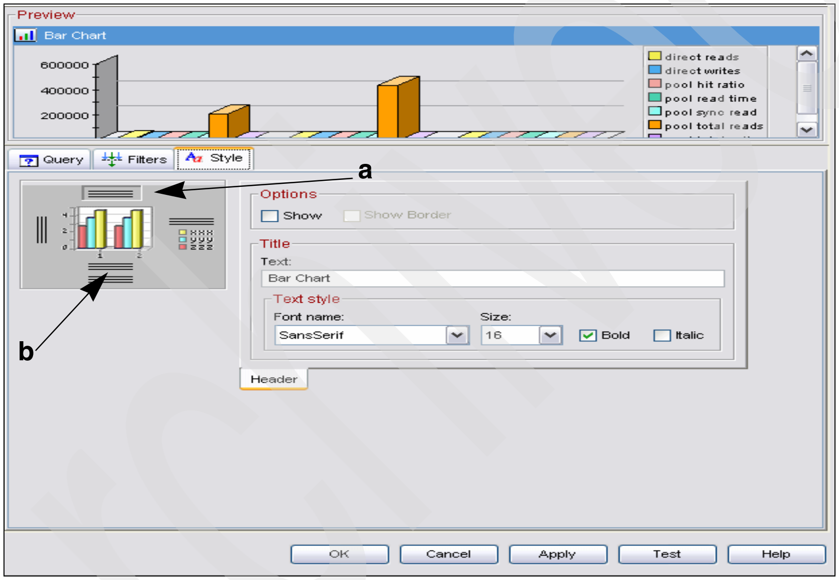


Figure 5-37 Default workspace: Using the Style tab

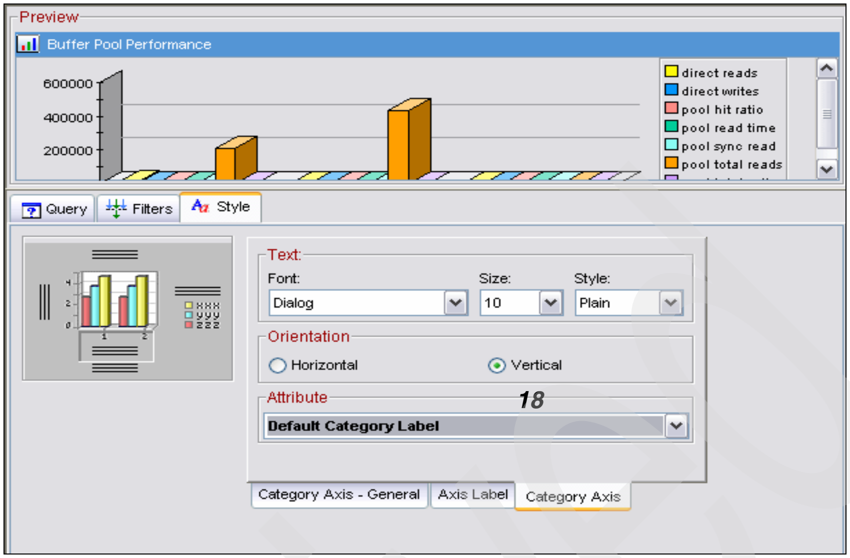


Figure 5-38 Default workspace: Change category axis

16.Click the Attribute list and select **bp name (Unicode)** (Figure 5-39).

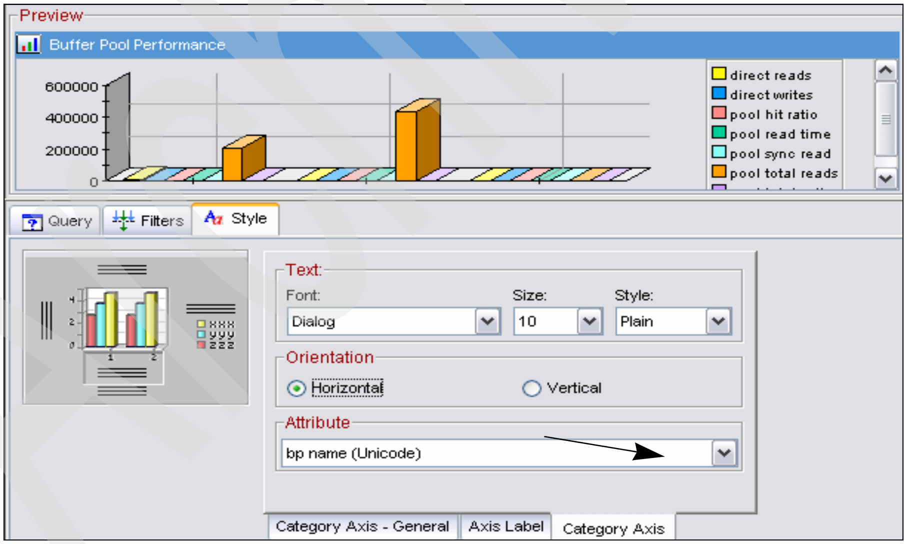


Figure 5-39 Default workspace: Select bp name

17.Click **OK**.

Your view should now look like the view Figure 5-40.

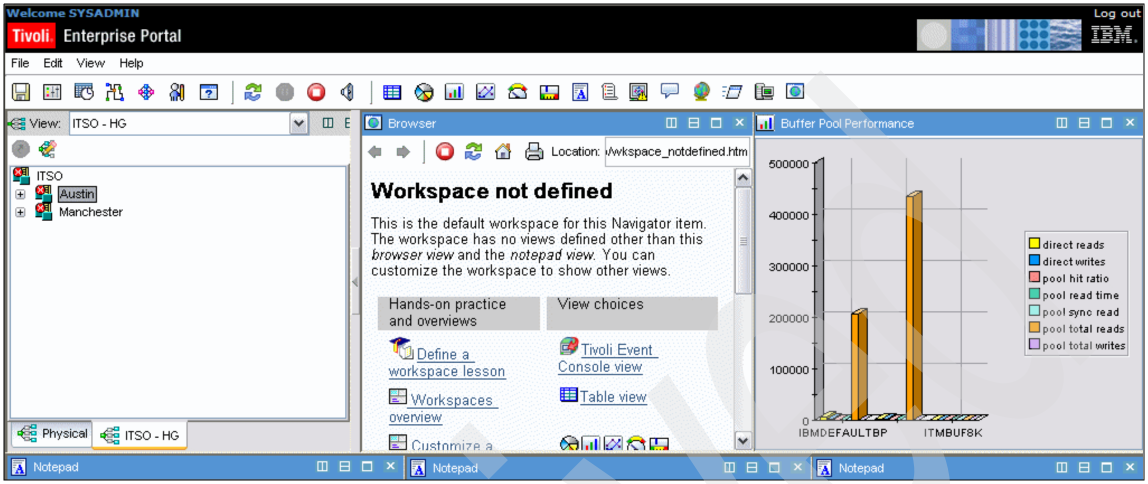


Figure 5-40 Default workspace with correct chart name and data

Repeat this process for the remaining panes in your workspace.

5.6 Monitoring the Tivoli environment

For purposes of having another real-life scenario, we explore monitoring the Tivoli infrastructure. In this example, we build some components to monitor our Tivoli environment.

Look purely at the availability of the Tivoli framework. Note that situations are platform-specific and you have to build similar situations for Windows and UNIX that will differ in the attributes used.

5.6.1 Monitoring the Tivoli Monitoring Express V6.1 servers

The IBM Tivoli Monitoring Express V6.1 infrastructure is very similar to IBM Tivoli Monitoring V6.1. It uses a hub Tivoli Enterprise Monitoring Server and a Tivoli Enterprise Portal Server. With IBM Tivoli Monitoring V6.1, you can also use a remote Tivoli Enterprise Monitoring Server for scalability. Although a remote Tivoli Enterprise Monitoring Server is not an option during the installation of IBM Tivoli Monitoring Express V6.1, you can install and configure a remote monitoring server if required. Because these components are critical to the monitoring solution, monitor their performance and availability, too.

Because there is just a heartbeat between the interconnected Tivoli Enterprise Monitoring Server systems, you merely need to access that information to obtain the status of all remote monitoring servers and portal servers from the hub monitoring server.

In this example, we create a new situation called ITSO_ITM_Offline_CRITICAL and selected Tivoli Enterprise Monitoring Server as our monitored application, as shown in Figure 5-41.

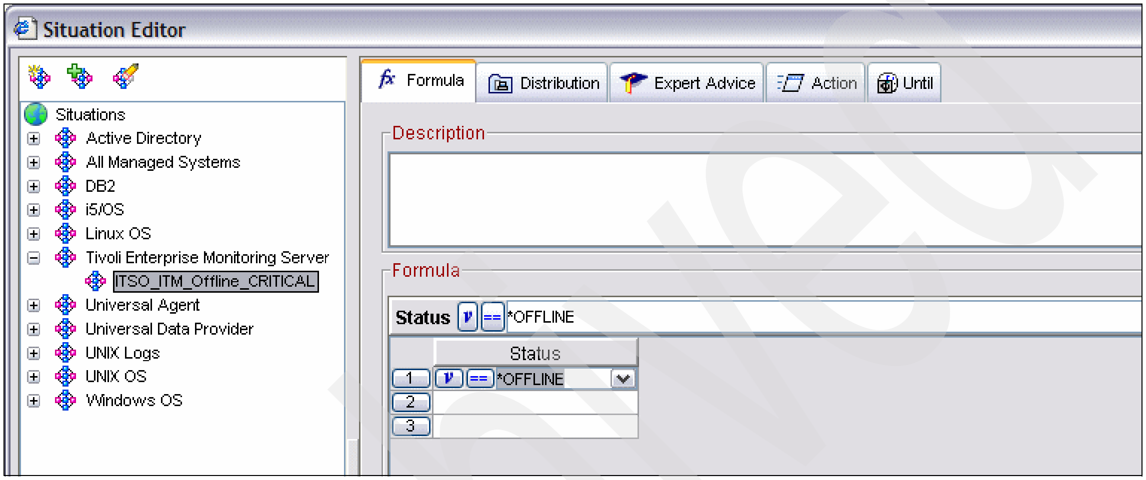


Figure 5-41 Building a situation to monitor Tivoli Monitoring Express components

This situation creates events to show that IBM Tivoli Monitoring Express V6.1 components are offline in our Tivoli Enterprise Monitoring Server environment.

5.6.2 Monitoring users logged in to Tivoli Monitoring Express V6.1

Another useful bit of information is a list of logged-in IBM Tivoli Monitoring Express V6.1 users. This information is contained in the Tivoli Enterprise Portal Server database. In exploring the method for obtaining this information, you can see how to query the information in any database.

First, identify the database information. In our example environment, the portal server uses a DB2 database. Using the control center on Windows, peruse the database from a DB2 perspective to see what kind of information is available. In this example, we see that the information needed is contained in the KFWLOGIN and KFWUSER tables in the DB2 database.

Create the query that will retrieve the information from the database. To launch the Query Editor from the Windows Tivoli Enterprise Portal, press Ctrl+Q. Name the query and select the appropriate data sources, as shown in Figure 5-42.

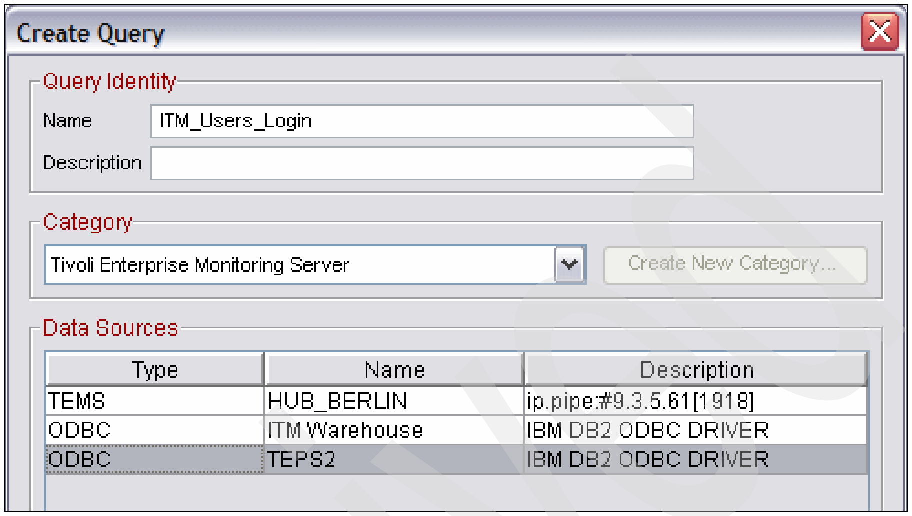


Figure 5-42 Defining the query for which users use Tivoli Enterprise Portal Server

After defining the query, provide the SQL for the query. In this example, we use `SELECT * FROM KFWLOGIN`, which basically returns the entire table. You can just as easily pare it down to an attribute or two or hide the columns when using this query to build a report view of this information. Running the query opens the window shown in Figure 5-43.

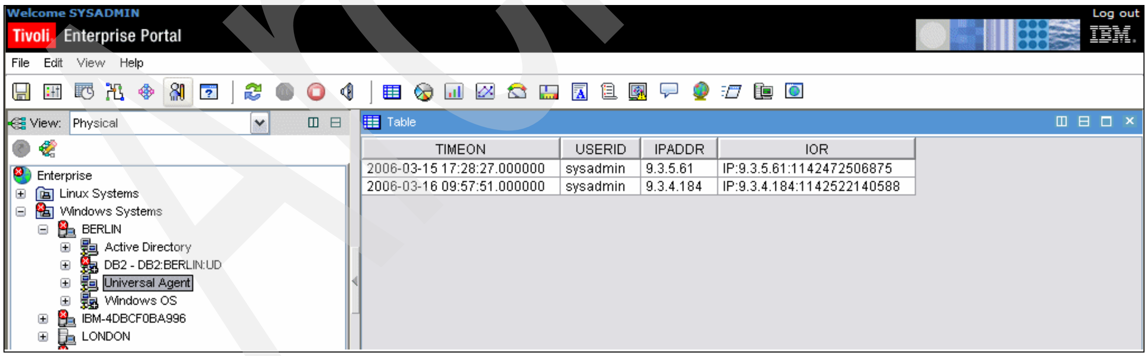


Figure 5-43 Result from running the query ITM_Users_Login

5.7 DB2 UDB scenarios

DB2 UDB environments can range from stand-alone systems to complex combinations of database servers and clients running on multiple platforms. In all, the common key for successful applications is performance. When you plan, design, and build your database system, you need to understand various considerations about the logical and physical database design, application design, and configuration parameters of DB2 so that your system can meet the performance requirements of your applications.

In this section, we discuss tasks involved in tuning the database and Windows environments to obtain optimal performance. We cover the major items that experience has shown have the largest impact on performance:

- Performance tuning overview
- Primary Windows performance factors
- Primary DB2 performance factors
- Using predefined situations to monitor a DB2 server
- Using custom situations to monitor a DB2 server

We also provide guidance about where to find more detailed information about specific performance issues using the IBM Tivoli Monitoring Express V6.1 solution and its monitoring and alerting capabilities.

5.7.1 Performance tuning overview

Although the performance of your database system might initially be good, as time goes on, it might need to server more users, store more data, and process more complex queries. Consequently, the increased load level of your database server will affect its performance. Some experienced people might say it is time to upgrade to more powerful equipment, but before investing in equipment, you might be able to improve performance by simply tuning the database system using IBM Tivoli Monitoring Express V6.1.

Performance is the capacity of your system to produce the desired results with a minimum cost of time or resources. It can be measured through response time, throughput, and availability.

Performance is not an absolute value. The performance of an information system can be rated as better or worse compared to a reference value. First, the reference value needs to be established according to the requirements of the information system; then, the results of tuning efforts can be compared against it. Those requirements, or the service level agreement, can include the throughput of the system, limits on the response time for a percentile or transactions, or any other issues relevant to the end user.

Units of measurement are usually based on the response time of a given workload. Other units of measurement can be based on transactions per second, I/O operations, CPU use, or a combination of these.

Your database system is a complex data-processing environment that includes hardware resources, software components, and application programs. DB2 starts many processes that perform different functions in your database system, and it allocates the necessary memory areas, thus consuming hardware resources.

As system performance degrades, you might first be tempted to upgrade your system with more powerful and expensive equipment. However, in the meantime, you might be able to improve the performance of your existing resources by simply tuning your operating system and databases. By carrying out a performance tuning project, you can balance your hardware resources to each part of the database system, including processes and the required memory areas. Specific goals of tuning can include:

- Processing a larger or more demanding workload without buying new hardware
- Obtaining faster response times, or higher throughput, without increasing processing costs
- Reducing processing costs without negatively affecting service to your users, and spending the money for other resources

Other benefits are intangible, for example, greater user satisfaction and productivity resulting from faster response times. If you manage an Internet business, higher performance, including quick response time and high availability, might prevent lost business opportunities. When weighing the cost of performance tuning against its possible benefits, all of these benefits need to be considered.

IBM Tivoli Monitoring Express V6.1 can kick-start your tuning exercise with its out-of-the-box provided workspaces. These workspaces will give you an indication of where the performance bottleneck might be. From any of these workspaces, you can navigate to another workspace for more detailed information about selected object, a database, application, tablespace, and so on.

See 5.7.3, “Primary DB2 performance factors” on page 259 for more information about these workspaces.

5.7.2 Primary Windows performance factors

We begin our discussion about system performance by taking into consideration the primary Microsoft Windows performance factors. As with any system, performance begins by laying a good foundation of well-balanced hardware resources that can be exploited by the operating system and eventually application-specific software such as DB2 UDB.

In this section, we focus on Windows performance factors system hardware for which IBM Tivoli Monitoring Express V6.1 can help monitor and integrate with overall DB2 UDB monitoring. Specifically, we cover:

- Memory
- Processor
- Storage
- Network

Memory

Because accessing data in memory is faster than accessing data on hard drives, the primary factor in terms of memory is quantity. Although memory speeds are also factors, it is seldom an option when configuring a system unless you are willing to select and configure another system altogether. The amount of physical memory is a critical system hardware resource that can have a huge impact on overall performance. In general, the cost of memory on commodity servers is usually an insignificant factor when compared to the cost of other hardware resources. Figure 5-44 demonstrates an example of a monitoring view of memory allocation.

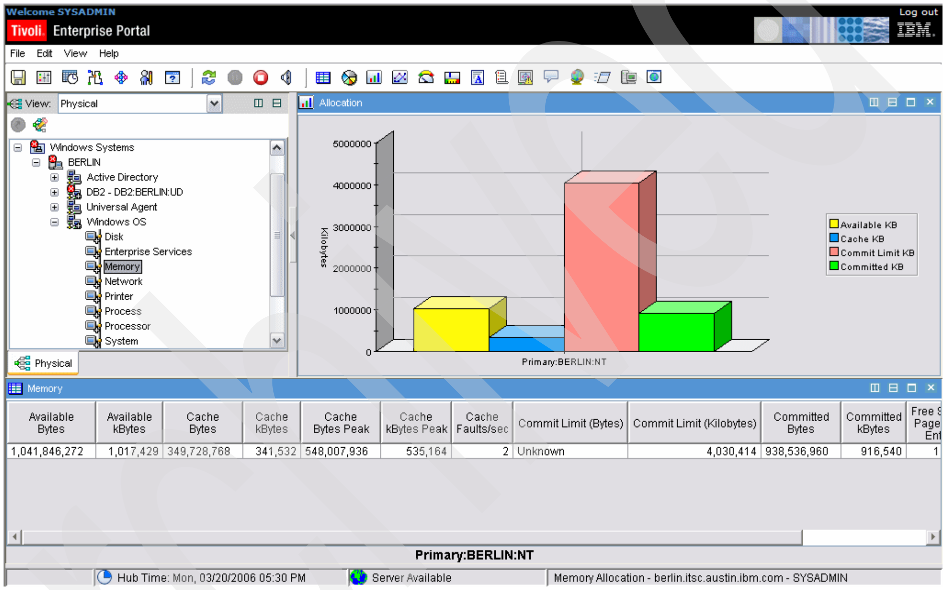


Figure 5-44 Windows: Memory allocation

Processor

Most systems are limited by the total number of central processing units (CPUs) they can support. Typically, a 4-way cannot be upgraded to an 8-way unless it is indeed a true 8-way that was populated with only four processors. In addition to quantity and speed, another important consideration in terms of processor selection is the size of the internal L2 cache. Slower processors with larger internal caches have shown significant throughput advantages for database applications over faster processors with smaller internal caches.

Figure 5-45 demonstrates a monitoring view of a Windows processor.

Another factor to consider when selecting the number of processors is the operating system software costs. There are incremental licensing costs associated with each Windows 2000 or Windows Server 2003 edition to support more processors.

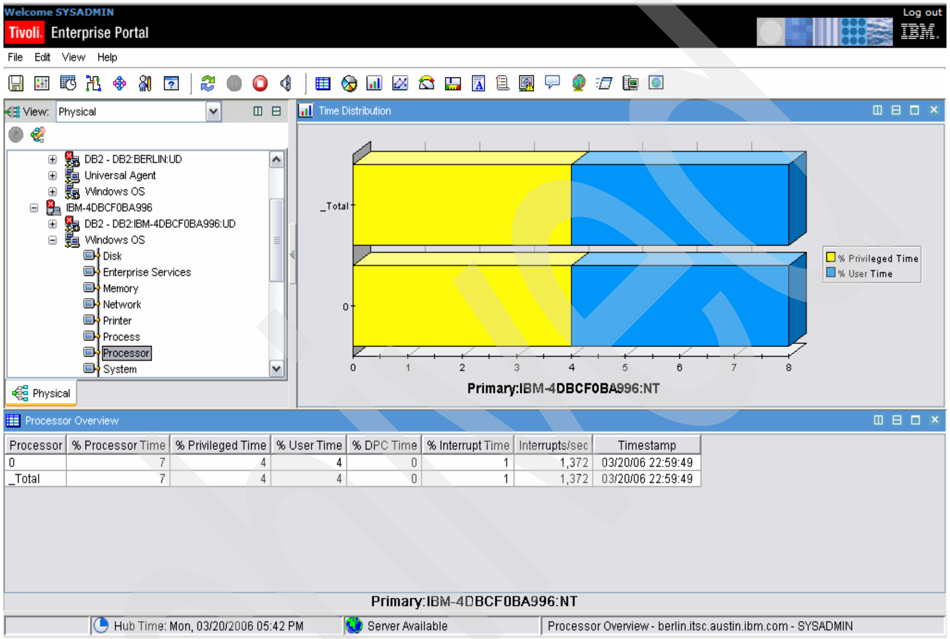


Figure 5-45 Windows: Processor overview

Storage

The disk subsystem has been an area of much debate over the last several years. Most disk subsystems will implement some form of redundancy that has always favored recoverability over performance. In recent years, improvements in technology has been able to overcome many of the performance limitations imposed by implementing redundant disk arrays.

Performance characteristics of disk controllers include speed, throughput, channels, and cache. Care should be taken in the placement of disk controllers in the system. Although most disk adapters are backward compatible, it should go without saying that you want to match the disk controller’s speed with that of the system’s PCI bus.

Performance characteristics of disk subsystems include disk speed, size, cache, and the number of physical disks in the subsystem. You should favor a subsystem with a large number of small drives over a small number of large drives. If this is impractical, plan for growth by choosing a large number of large drives. Best performance will be achieved for database applications with a large number of physical disks (5-10) per processor.

Figure 5-46 shows an example of a monitoring view for storage.

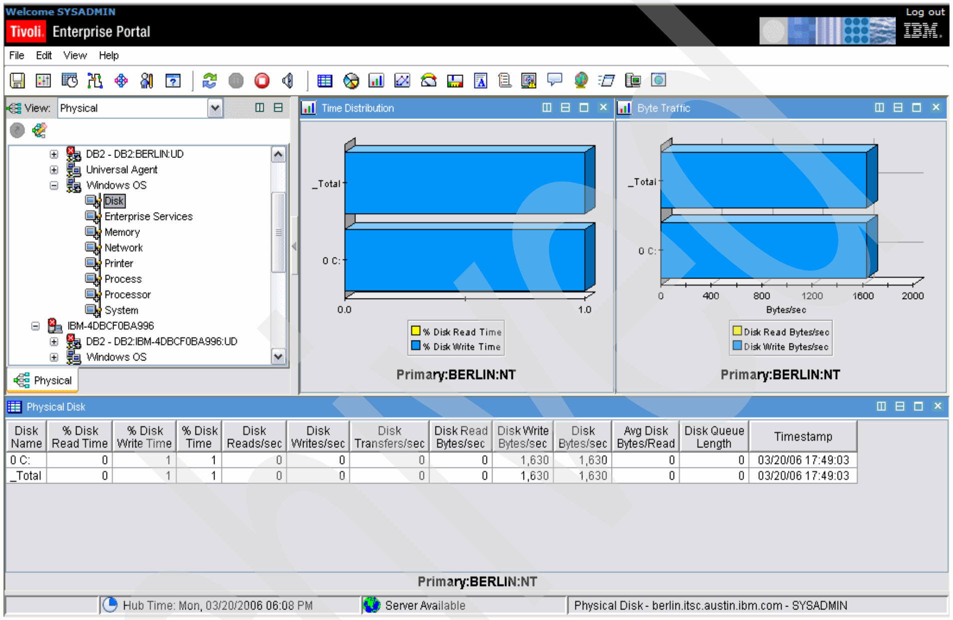


Figure 5-46 Windows: Storage overview

Network

Performance characteristics of network adapters include speed and throughput. As with disk controllers, care should be taken in the placement of network adapters in the system. You should also consider the number of network adapters in your system. If possible, also avoid placing network adapters on PCI buses populated with disk controllers.

The speed of the network adapter can limit the total network throughput. Consider using faster 64-bit 66 MHz network adapters, especially when running on gigabit networks as slower 32-bit 33 MHz network adapters are not capable of driving gigabit networks.

Most network adapter today support teaming. Teaming network adapters provides several benefits. First, it provides network redundancy by preventing a single network adapter that can be a single point of failure in your system.

Second, it provides better performance because you can balance network traffic over two or more adapters and PCI buses.

The workspaces we use in this section can also be consolidated into one view so that a DBA together with a Windows administrator can get an overall view of how these hardware components are performing (Figure 5-47).

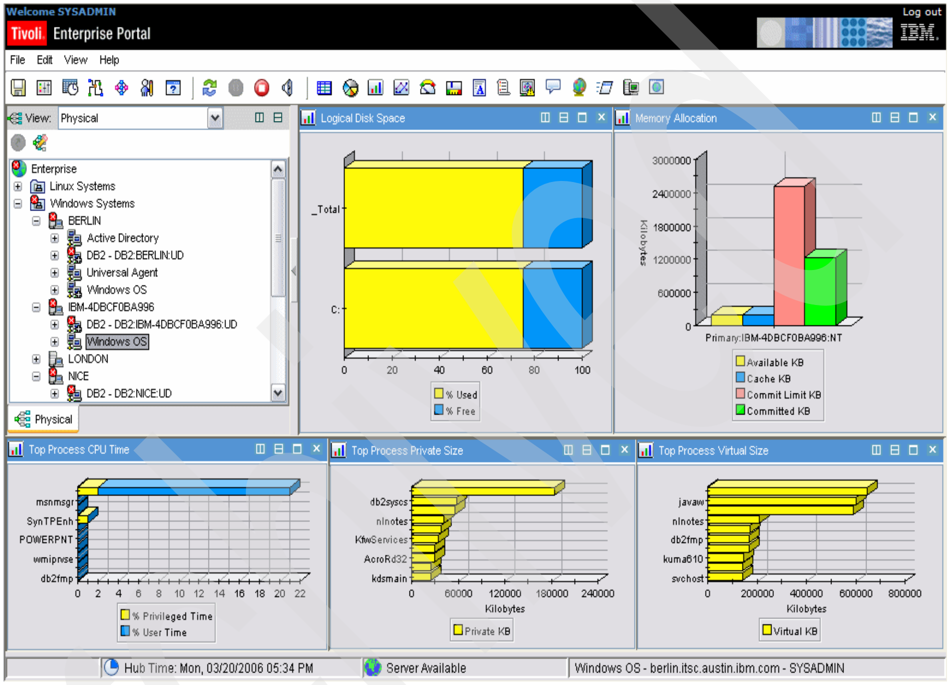


Figure 5-47 Windows systems overview

5.7.3 Primary DB2 performance factors

With IBM Tivoli Monitoring Express V6.1 comes a number of predefined workspaces. In this section, we discuss the following scenarios in more detail:

- Application top 10 summary workspace
- Buffer pool workspace
- Database overview workspace

Application top 10 summary workspace

The LOCKTIMEOUT configuration parameter specifies the number of seconds that an application waits to obtain a lock. By specifying a maximum value, you can avoid global deadlocks for applications. If you set this parameter to 0, an application does not wait for a lock. In this case, if a lock is not available at the time of the request, the application receives notification of this immediately. If you set this parameter to -1, lock timeout detection is turned off. In this case, an application waits for a lock (if one is not available at the time of the request) until the lock is granted or a deadlock occurs.

Set the value to detect quickly any waits that are occurring because of an abnormal situation, such as a transaction that is stalled. Set the LOCKTIMEOUT parameter high enough that valid lock requests do not time out because of peak workloads. However, the LOCKTIMEOUT configuration parameter can be set too high, which causes the system to experience too few lock timeouts. In this case, applications might wait excessively to obtain a lock (Figure 5-48).

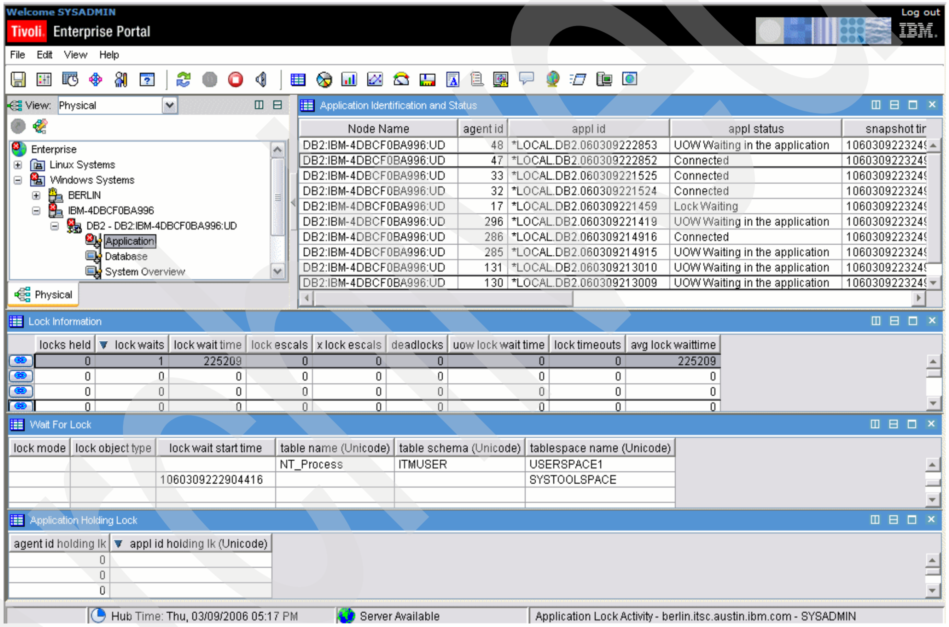


Figure 5-48 Application top 10 summary workspace: Locking activity

You can use the application top summary workspace to help track the number of times an application (connection) experienced a lock timeout. By using the application lock activity workspace, you can view the Lock_Timeout attribute in addition to other lock-related attributes. This attribute indicates the number of times that a request to lock an object timed out instead of being granted. A high value for the Lock_Timeout attribute can be caused by:

- The value of the LOCKTIMEOUT configuration parameter being too low
- An application (transaction) that is holding one or more locks for an extended period

- A concurrency problem that can be caused by lock escalations (from the row level to a table level)

The Lock_Timeout attribute can help you adjust the setting for the LOCKTIMEOUT configuration parameter. If the number of lock timeouts becomes excessive when compared to normal operating levels, you might have an application that is holding locks for long durations. In this case, this attribute might indicate that you need to analyze some of the other attributes related to locks and deadlocks to determine if you have an application problem.

Online transaction processing (OLTP) applications should not perform large sort operations. Large sort operations are very costly in terms of CPU, I/O, and elapsed time; as a result, sorts can slow down an OLTP application. The default SORTHEAP size is 1 MB (256 4-KB pages), which is adequate for most situations. You can use the information in the application top summary workspace to help you track the number of sort overflows.

In the application summary workspace, you can view information about the number of sort overflows and the sort overflow percentage. Additionally, you can use the application sort and hash join activity workspace (Figure 5-49) to find information about the total number of sorts, the average sort time, the number of sort overflows, and the percentage of sorts that cause an overflow condition. Sort overflows indicate that large sorts are occurring. If the number of sort overflows represents greater than 3% of the sorts, an application might experience serious, unexpected sort problems. You must identify the SQL statements that are causing the sorts and modify the SQL, indexes, or clustering to reduce the cost of the sorts.

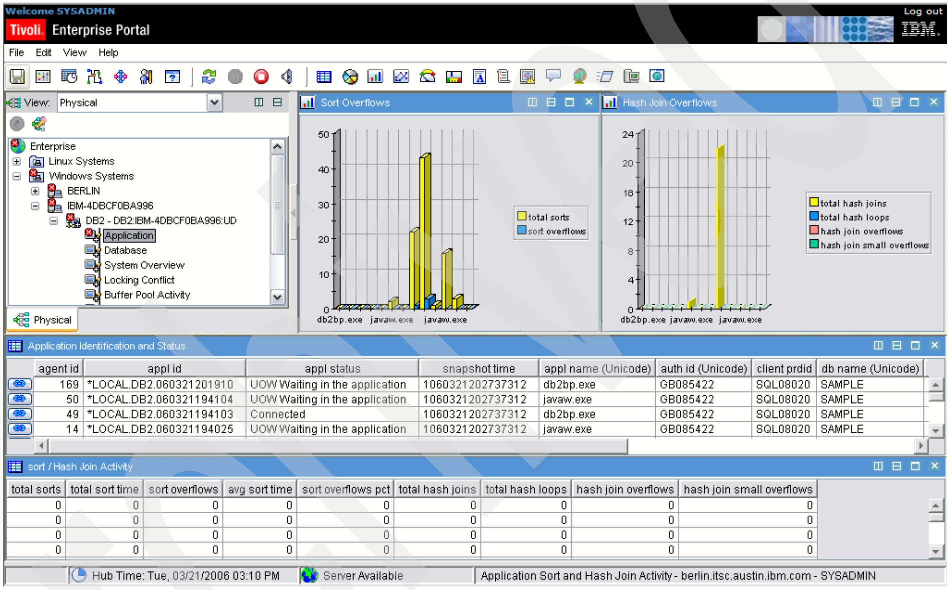


Figure 5-49 Application top 10 summary workspace: Sort and hash join activity

Buffer pool workspace

Database performance and tuning always start with buffer pool efficiency. The buffer pool hit ratio indicates the percentage of time that the database manager did not need to load a page from disk in order to satisfy a page request. That is, the page was already in the buffer pool. The greater the buffer pool hit ratio, the lower the frequency of disk I/O. If the buffer pool hit ratio is low, the database will experience excessive I/O activity. If this is the case, consider enlarging the buffer pool size for frequently accessed tables or placing the indexes into a separate buffer pool. Buffer pools that are too small result in excessive, unnecessary, physical I/O. Buffer pools that are too large put a system at risk for operating system paging activity. Figure 5-50 provides an overview of the buffer pool activity.

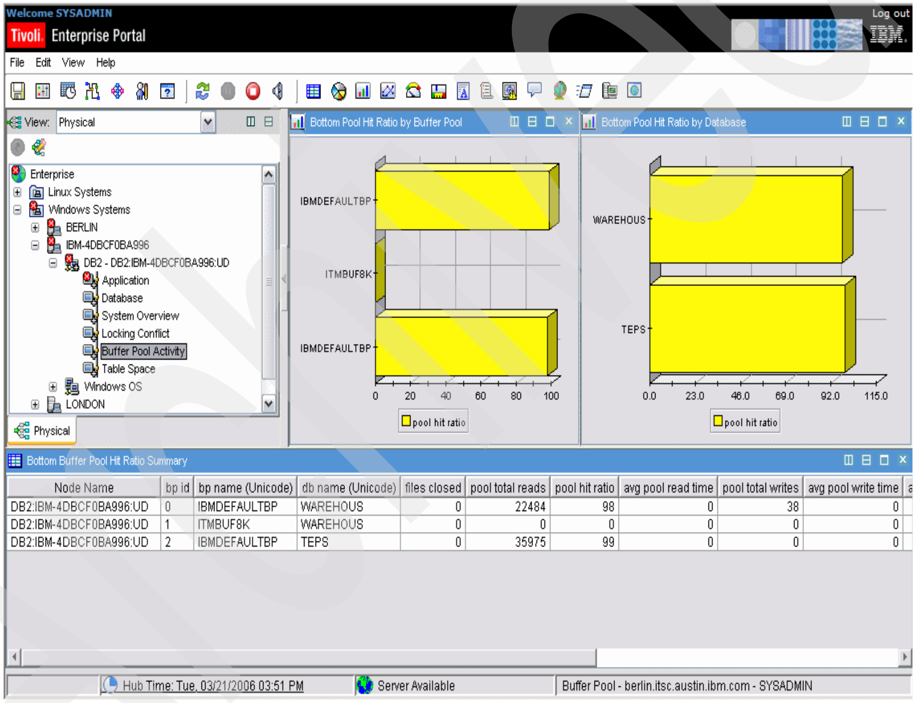


Figure 5-50 Buffer pool summary workspace

You can use the information displayed in the buffer pool workspace to evaluate many of the characteristics of buffer pool activity. In the associated buffer pool detail workspace (Figure 5-51), you can evaluate the values of the various attributes related to buffer pool hit ratios, asynchronous and synchronous I/O activity, and extended store and non-buffer pool I/O activity. With this information, you can identify aspects of buffer pool activity that are outside normal operating levels and take corrective action.

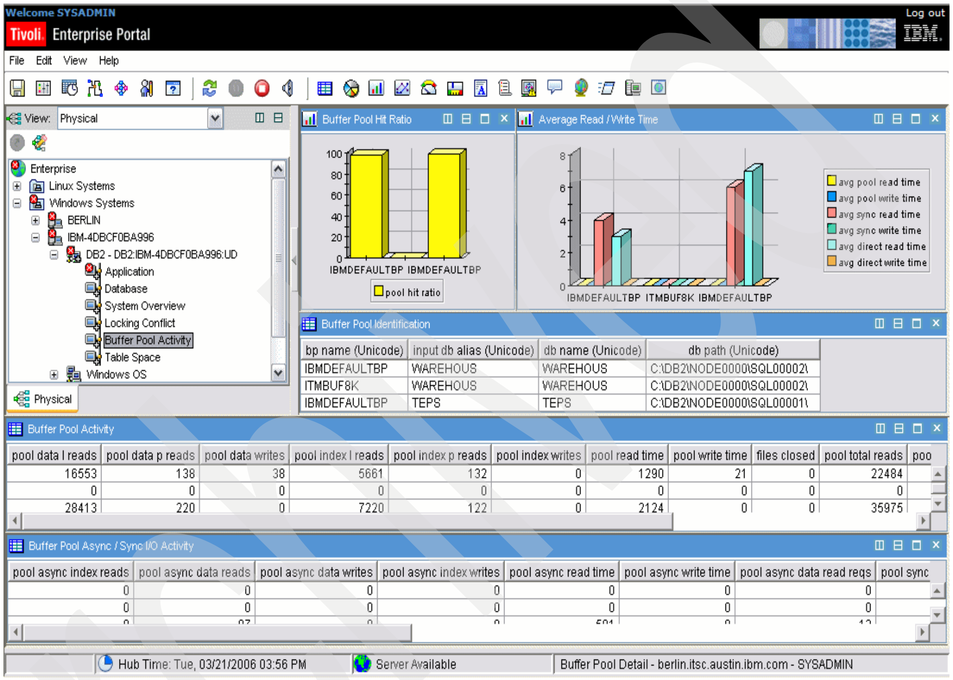


Figure 5-51 Buffer pool detail workspace

Database overview workspace

The MAXFILOP parameter specifies the maximum number of database files that any single database agent can have open at the same time. If opening a file might cause this value to be exceeded, DB2 closes a file already in use by this agent. If the value of MAXFILOP is too small, DB2 encounters increased processor usage for opening and closing files so that the system does not exceed this limit. The processor usage can become excessive and cause performance degradation. SQL response time can slow considerably. You can monitor the opening and closing of files by using the database overview workspace, database summary workspace, and database I/O activity workspace (see Figure 5-52 on page 265).

In the Buffer Pool Activity area, you can determine the value of the Files Closed attribute.

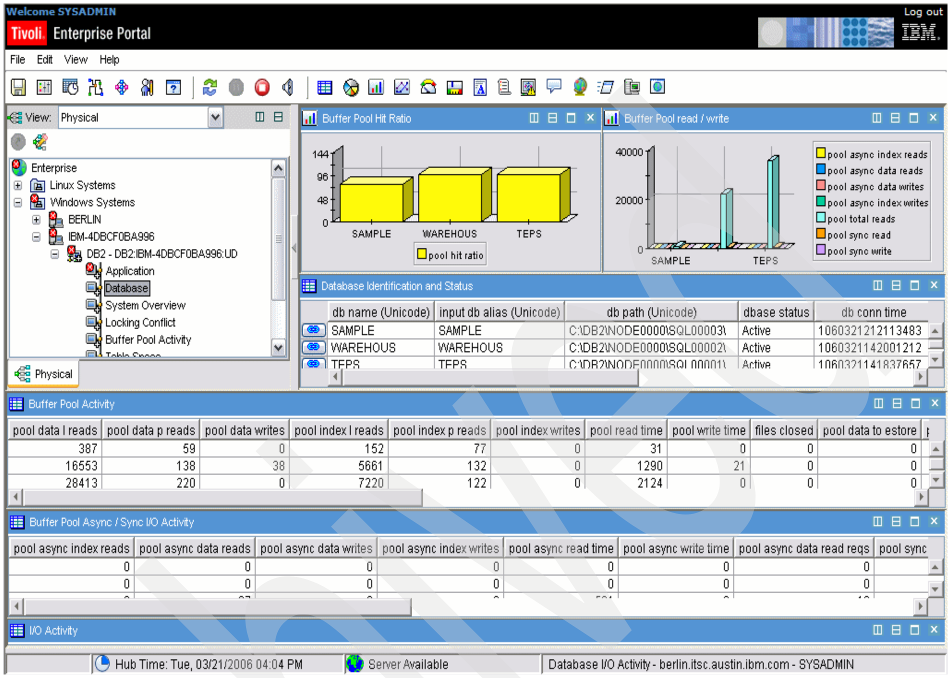


Figure 5-52 Database I/O activity workspace

The Files Closed attribute can help you determine the best value for the MAXFILOP configuration parameter. If the number of files being closed exceeds the norm in your environment, consider increasing the value of the MAXFILOP parameter until the opening and closing reaches an acceptable level.

Another frustrating tasks that DBAs have to perform is trying to find out when and why one database transaction is blocking another transaction from performing. Although most databases have built-in deadlock detectors and timeouts to allow these sorts of issues to be resolved, being able to easily identify applications or requests that block others can significantly enhance the end-user experience of your system with minimum, manual intervention.

The IBM Tivoli Monitoring Express V6.1 for Databases solution (including DB2 UDB, Oracle, and Microsoft SQL Server) helps you improve the efficiency of database performance monitoring, problem determination, and resolution. By tracking a set of predefined monitor data used in situations, the solution enables you to quickly locate the cause of the problem. You can then take direct action to resolve the problem.

The solution also help you monitor application performance, application concurrency, resource consumption, and SQL statement usage. The solution will assist you in diagnosing database performance problems such as lock-waiting situations and in tuning queries for optimal utilization of the database resources.

So what we need is a custom Navigator view that will provide us with an overview of locking activity from an application and database perspective.

Figure 5-53 shows an example of such a view.

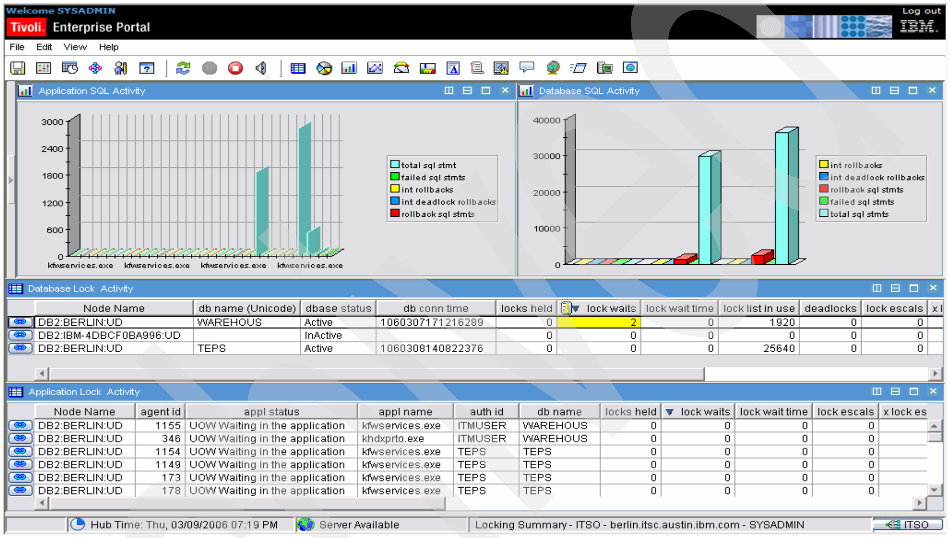


Figure 5-53 Locking overview

For information about how to build a custom view like this, see 5.5.1, “Building a hierarchy of Navigator views” on page 234.

5.7.4 Using predefined situations to monitor a DB2 server

You can display predefined situations using the Situation Editor in the Tivoli Enterprise Portal client. The left frame of the Situation Editor initially lists the situations associated with the Navigator item that you selected.

Table 5-1 is a list of best practice predefined situations. These are the situations that are best suited for the needs of small and medium businesses. If necessary, you can change the conditions or values being monitored by a predefined situation to those best suited to your client's business.

Table 5-1 Best practice predefined situations for DB2

Predefined situations	**Description**
UDB_Appl_BP_Hit_Ratio_Low	This situation monitors the buffer pool hit ratio for an application. It triggers a warning alert if an application experiences a buffer pool hit ratio below 50%. This alert warns of a badly performing application. Repeated warnings of this type indicate a need for system tuning.
UDB_Appl_CatCache_Hit_Low	This situation issues a warning alert if an application experiences a catalog cache hit ratio that is lower than 50%.
UDB_Appl_Lock_Warning	This situation monitors an application for one or more of the following conditions: ► More than 5 deadlocks ► More than 5 lock timeouts ► More than 20 lock waits It triggers a warning alert if an application experiences one or more of these conditions.
UDB_BP_Hit_Ratio_Low	This situation monitors the buffer pool hit ratio. It issues a warning alert if the buffer pool hit ratio is below 50%.
UDB_Database_Lock_Warning	This situation monitors the database for the following conditions: ► More than 10 deadlocks ► More than 10 lock timeouts ► More than 20 lock waits It issues a warning alert if the monitored database experiences these conditions. This is useful in system/application tuning. The more time that elapses after system reset before one of these events occur, the better.
UDB_DB_BP_Hit_Ratio_Low	This situation issues a warning alert if a database's buffer pool hit ratio falls below 65%. This situation is based on the hit ratio during the monitoring interval and can vary widely depending on the applications.
UDB_DB_Cur_Cons_Pct_Warn	This situation issues a critical alert if the percentage of connections used exceeds the critical threshold. It warns of databases nearing their maximum connection limit.
UDB_DB_Dlk_Rb_Pct_For_Int_Warn	This situation issues a warning alert if the internal deadlock rollbacks percent during the monitoring for interval exceeds the critical threshold.

Predefined situations	Description
UDB_DB_Lock_Waits_Pct_Crit	This situation issues a critical alert if the percentage of applications in a lock wait state exceeds the critical threshold.
UDB_DB_Pool_Hit_Ratio_Pct_Crit	This situation issues a critical alert if the percentage buffer pool hit ratio (data plus index) exceeds the critical threshold. Also, consider using UDB_DB_Pool_Hit_Ratio_Pct_Warn, which is the same situation but with a warning alert.
UDB_DB_Pri_Log_Used_Pct_Crit	This situation issues a critical alert if the percentage used in the primary log exceeds the critical threshold.
UDB_DB_Sec_Log_Used_Pct_Crit	This situation issues a critical alert when the percentage used in the secondary log exceeds the critical threshold.
UDB_DB_SQL_Fail_High	This situation monitors SQL statement failures. It issues a warning alert if a monitored database experiences more than 40% SQL statement failures.
UDB_Post_Threshold_Sort_High	This situation issues a warning alert if the UDB server experiences more than 20 post-threshold sorts.
UDB_Status_Warning	This situation issues a warning alert if the status of the monitored UDB instance is other than active status.
UDB_TS_Sp_Used_DMS_Tab_Pct_Crit and warn	This situation issues a critical alert if the percentage of space used in the DMS tablespace exceeds the critical threshold.

5.7.5 Using custom situations to monitor a DB2 server

When your environment requires situations with values that are different from those in the predefined situations or when you need to monitor conditions not defined by the existing situations, you can create custom situations to detect problems with resources.

Table 5-2 shows custom situations that are suggested for additional monitoring of IBM DB2 servers. You can create more custom situations depending on your client's environment requirements.

Table 5-2 Custom situations to monitor a DB2 server

Custom attributes	Description
Deadlocks for Int	This attribute is the number of deadlocks that occurred in the database during the monitoring interval and is in the KUDDBASE01 attribute group. A similar attribute exists in the KUDDB2APPLGROUP01 group; however, this attribute is at the application level rather than database. A situation based on this attribute can be useful for detecting a spike in deadlocks, perhaps caused by a change in one or more applications. The threshold depends the monitoring interval; however, a deadlock should be a rare event in a well-tuned database, certainly not more than 2 or 3 per hour.
Lock escalations for Int	This attribute is the number of lock escalations that occurred on any given application during the monitoring interval. It is in the KUDDB2APPLGROUP01 attribute group. Lock escalations reduce the ability of the system to run applications concurrently. We suggest that you start with a low threshold value and try tuning the system, particularly the LOCKLIST and MAXLOCKS parameters before increasing the threshold.

Figure 5-54 displays a custom developed situation checking the number of deadlocks that occur in a database. The figure shows the formula and settings for the custom ITSO_DB__Deadlocks situation.

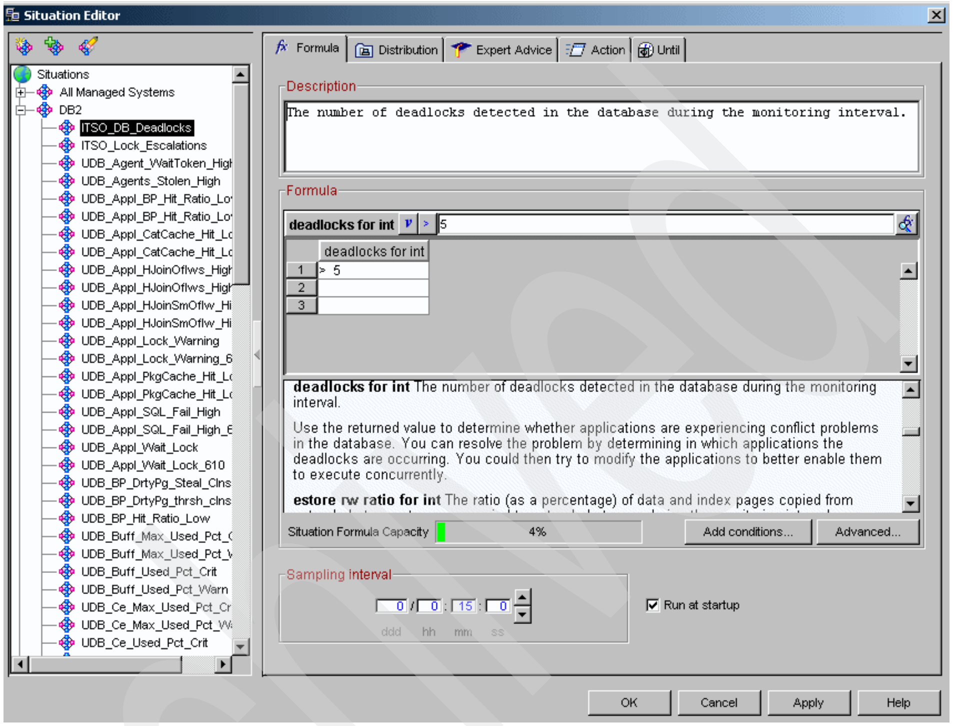


Figure 5-54 Custom situation in the Tivoli Enterprise Portal Situation Editor

Figure 5-55 shows another custom developed situation checking the number of lock escalations that occur in a database. The figure shows the formula and settings for the custom ITSO_Lock_Escations situation.

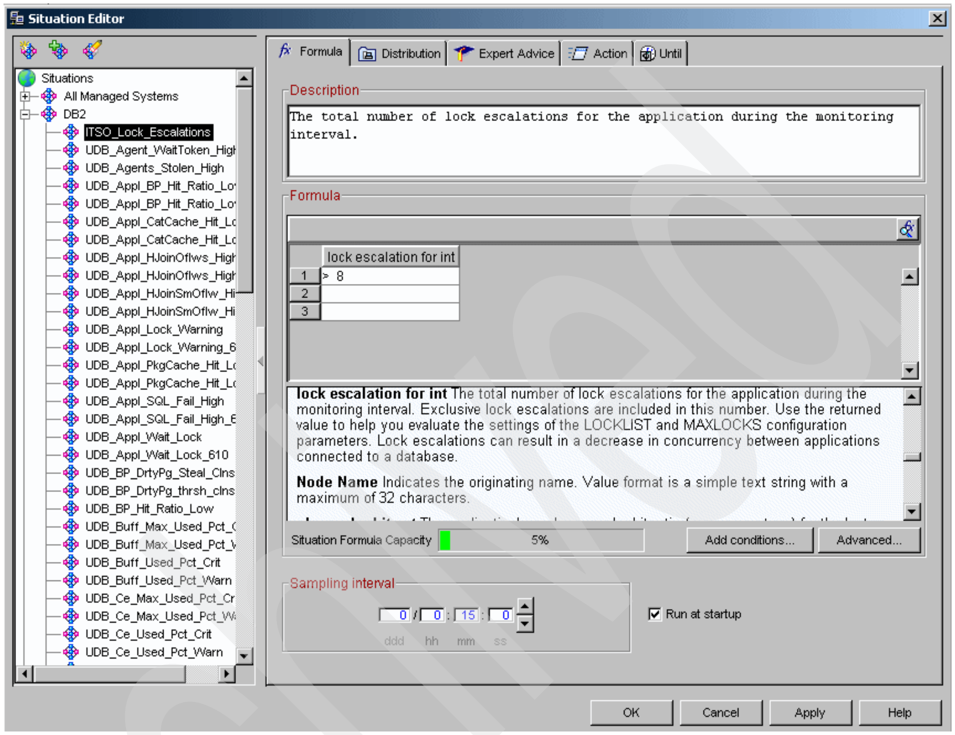


Figure 5-55 Custom situation in the Tivoli Enterprise Portal Situation Editor

5.8 Windows Active Directory scenarios

Microsoft Active Directory is a directory service. The term directory service refers to two things: a directory where information about users and resources is stored and a service or services that let you access and manipulate those resources. Active Directory is a way to manage all elements of your network, including computers, groups, users, domains, security policies, and any type of user-defined objects.

It melds several Windows NT® services and tools that have functioned separately so far (User Manager for Domains, Server Manager, Domain Name Server) and provides additional functions beyond these services and tools.

Active Directory is built around Domain Name System (DNS) and Lightweight Directory Access Protocol (LDAP), DNS because it is the standard on the Internet and is familiar, LDAP because most vendors support it. Active Directory clients use DNS and LDAP to locate and access any type of resource on the network.

The two most important goals of Active Directory are:

- Users should be able to access resources throughout the domain using a single logon.
- Administrators should be able to centrally manage both users and resources.

Monitoring Windows Active Directory performance is vital to making sure that Active Directory is meeting your business and networking goals. For example, one aspect of ensuring optimal performance is to verify that all network servers are getting directory replication updates and applying them in a timely manner. To monitor replication, as well as other activities, you have available a variety of predefined workspaces out-of-the-box with IBM Tivoli Monitoring Express V6.1.

For example, if you need to discover whether a server is receiving directory replication updates and applying the updates in a timely fashion, you can select the replication workspace and view the current activity.

You can use the same workspace to ensure a timely replication of all network servers by looking at the DRA Pending Replication Synchronization attribute (Figure 5-56) to check the number of directory synchronizations that are queued for a server but not yet processed.

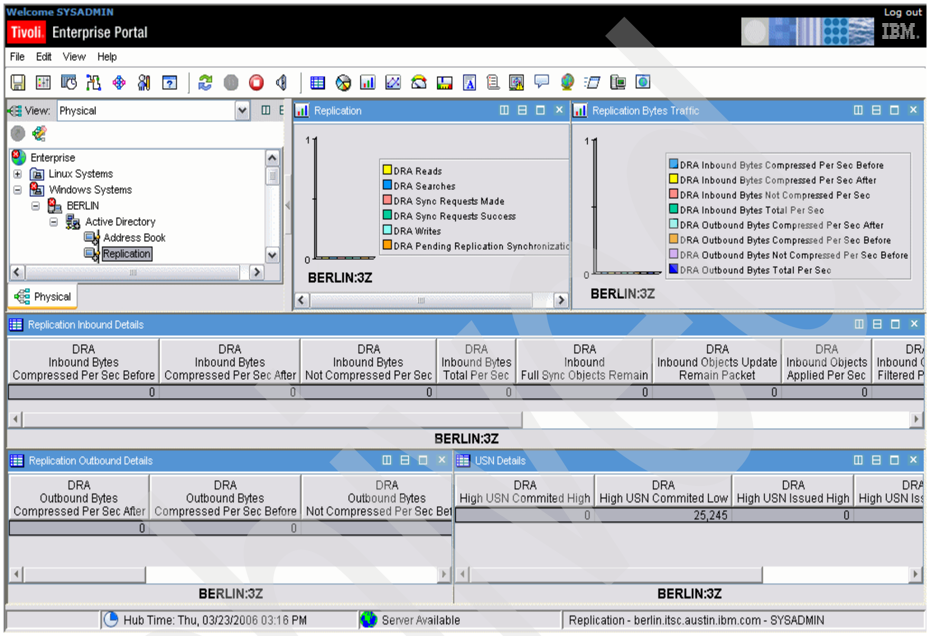


Figure 5-56 Windows Active Directory: Replication workspace

By using these workspaces and situations, you can monitor many activities in Active Directory. For example, activities such as monitoring replication topology, Domain Name System (DNS) functionality, latency, connection times, and allocation of relative identifiers (RIDs) can each be monitored by using IBM Tivoli Monitoring Express V6.1.

Tivoli Monitoring Express enables you to monitor the performance of local and remote computers anywhere in your network and summarize the performance at selected intervals. The collected data can also be stored in the Tivoli Data Warehouse component so that you can analyze the performance history of a computer.

By using Tivoli Monitoring Express V6.1, you can track the activity of performance objects through the use of workspaces (data) and situations.

Another important component to monitor is memory usage of the Local Security Authority. We can combine monitoring of memory usage and other important metrics regarding Local Security Authority using the same workspace. To do so, we need to add a query from the Windows OS attribute groups. Perform the following steps:

1. Select the **Local Security Authority** workspace and split the top graph and change the graph to a table (Figure 5-57).

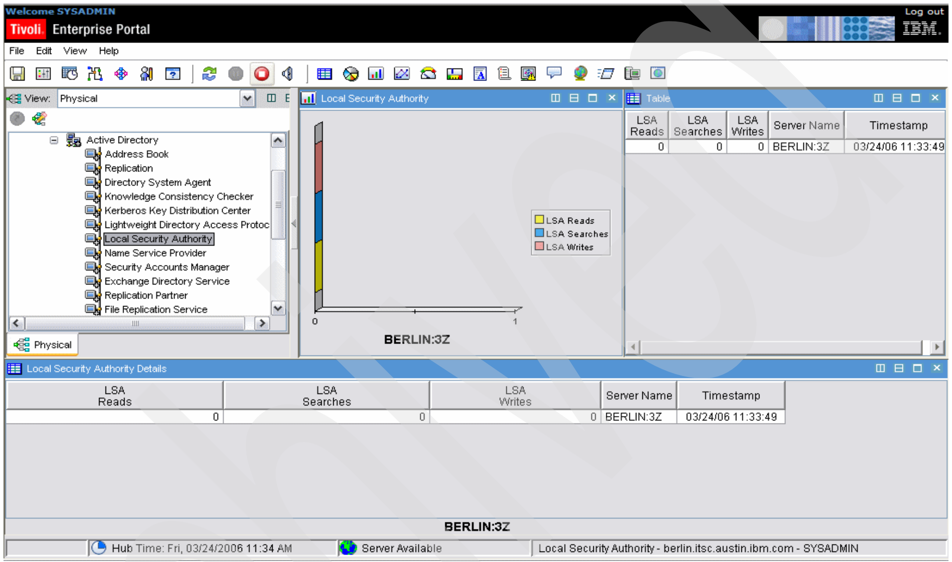


Figure 5-57 Local Security Authority workspace

2. Right-click the table on the right side and select **Properties**.
3. Click **Click here to assign a query** and expand Windows OS on the left side.

4. Click the plus sign next to NT Process and select the **Process Storage** query (Figure 5-58).

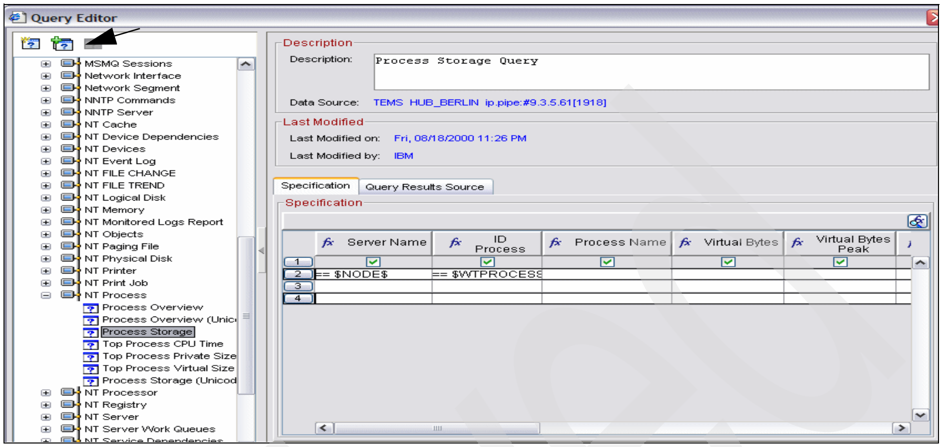


Figure 5-58 Query Editor: Process Storage

5. Because we will monitor a specific process, this query needs to be modified. Click **Create Another Query** on the top left corner (see arrow in Figure 5-58).

6. We give the query the name `LSA Process Memory`.

7. On the next panel (Figure 5-59), we enter the name of the process we want to monitor. In our scenario, we enter `'lsass'`.

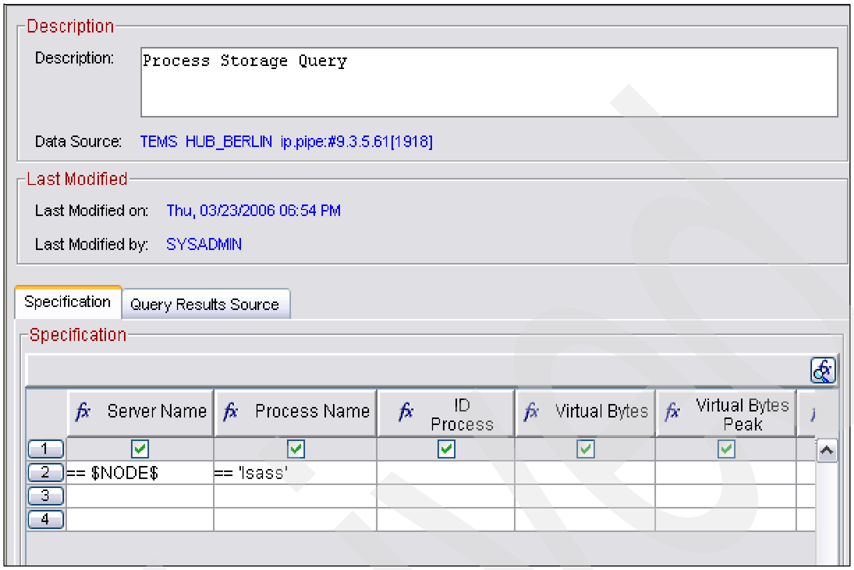


Figure 5-59 Enter name of process to monitor

8. Clear **ID Process**, because we are only interested in the name.
9. Before closing the panel, click **Query Results Source** (Figure 5-60).

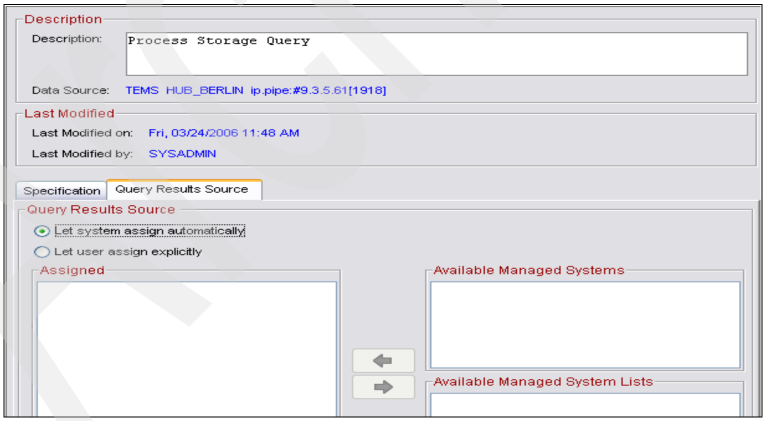


Figure 5-60 Query Results Source tab

10. As we can see in Figure 5-60 on page 276, there are no systems assigned to this query. To assign a system (agent), we click **Let user assign explicitly**.

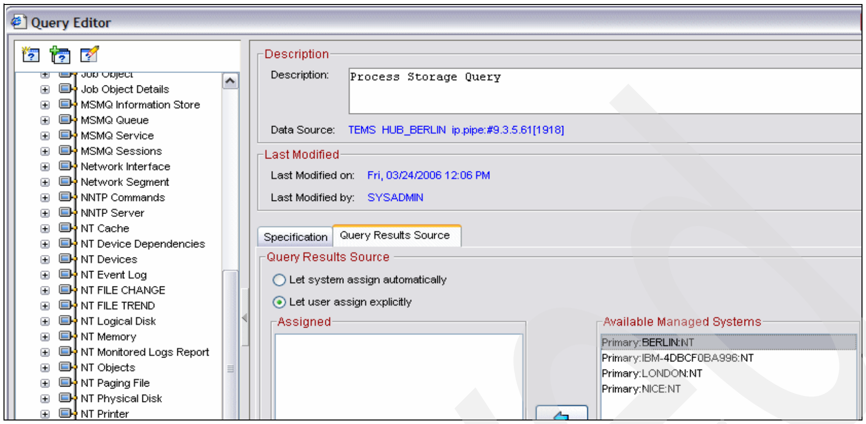


Figure 5-61 Select managed system

11. Now we can select the system, which in our scenario is Primary:BERLIN:NT, the server on which Active Directory runs. Select the name and click the blue arrow so that the name appears on the left side.

12. Click **OK**, and **OK** again.

13. You should now have a workspace that looks similar to the one shown in Figure 5-62.

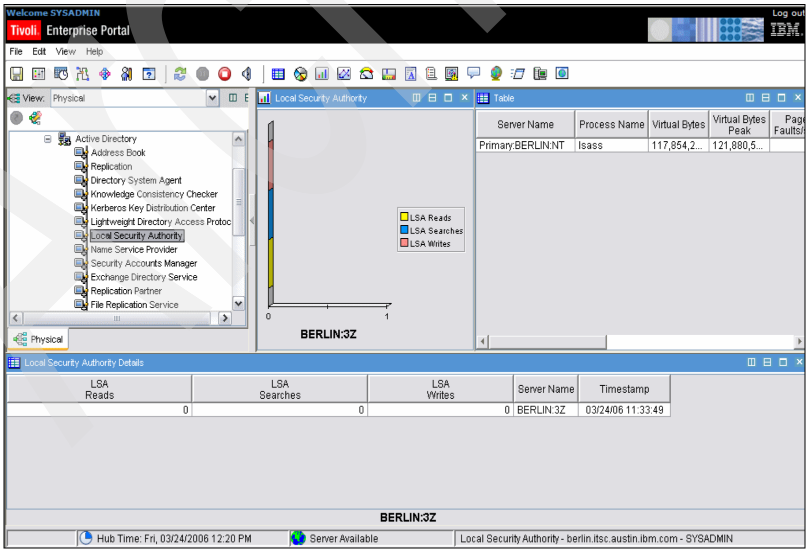


Figure 5-62 Local Security Authority and process memory data in the same workspace

14. Select **File** → **Save Workspace As** and give it a name that describes the content on the workspace.

Predefined situations for Monitoring Agent for Active Directory

Table 5-3 identifies the predefined situations that are best suited for the needs of small and medium businesses. If necessary, you can change the conditions or values being monitored by a predefined situation to those best suited to your client's business.

Table 5-3 Predefined situations for the Monitoring Agent for Active Directory

Predefined situations	Description
DHCP_Service_State_Critical	This situation monitors the availability of the DHCP services that might be critical to the operation of the Active Directory. A critical alert is triggered if the service becomes unavailable.
DNS_Service_State_Critical	This situation monitors the availability of the DNS services that might be critical to the operation of the Active Directory. It triggers a critical alert if the service is unavailable.
DNSAD_DC_SRV_Records_Bad_Warn	This situation triggers a warning alert when the copy of the zone that is stored on the specified server contains an SRV record for a domain controller that does not correspond to any of the known domain controllers that serve the domain covered by this zone.
DNSAD_DC_SRV_Recs_Missing_Warn	This situation detects when one of the domain controller SRV records is missing from the copy of the zone that is stored on the specified server and triggers a warning alert.
DNSAD_GC_SRV_Records_Bad_Warn	This situation detects when the copy of the zone that is stored on the specified server contains an SRV record for a global catalog that does not correspond with any of the known global catalogs that serve the forest. It triggers a warning situation.
DNSAD_GC_SRV_Recs_Missing_Warn	This situation monitors global catalog SRV records and triggers a warning alert, if necessary.
DNSAD_Node_Records_Missing_Crit	This situation monitors the DNS server for missing SRV records and triggers a critical alert, if necessary.

Predefined situations	Description
DNSAD_PDC_SRV_Records_Bad_Warn	This situation detects when the copy of the zone that is stored on the specified server contains an SRV record for a primary domain controller that does not correspond with the known primary domain controller that serves a specified domain. It triggers a warning situation.
DC_FSMO_Server_State_Critical	This situation monitors key services of a domain controller that holds an FSMO master role for Active Directory health.
DC_Server_FSMO_Status_Critical	This situation monitors key services of a domain controller that holds an FSMO master role for Active Directory health.
DCPerf_Cache_Page_Stalls_Warn	This situation monitors the number of page faults per second that cannot be serviced because there are no pages available for allocation from the database cache.
DCPerf_DB_Tab_Cache_Size_Warn	This situation monitors table hit statistics to determine if the ESE database table cache size is functionally too small.

5.9 Apache on Linux scenarios

IBM Tivoli Monitoring Express V6.1 does not have a unique agent for monitoring Apache. However, it does supply a generic agent called the Universal Agent for processing unsupported agent types. In this example, we use the IBM Tivoli Universal Agent to monitor Apache on Linux. With the Universal Agent, you can view the data in real time and historical workspaces on the Tivoli Enterprise Portal and manage with Tivoli Enterprise Portal monitoring situations and automation policies, the same as data from other Tivoli Enterprise Monitoring Agents.

5.9.1 Apache monitoring solution using the Universal Agent

The IBM Tivoli Universal Agent gets its data from interfaces called data providers. There are different types of data providers. For detailed information, refer to the Tivoli Monitoring V6.1.0 product Information Center.

http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/index.jsp?topic=/com.ibm.itm.doc/610uausers79.htm

IBM has created a special portal for IBM Business Partners and client called the IBM Tivoli Open Process Automation Library that provides extensions for IBM Tivoli Service Management applications. In this example, we use a downloadable Universal Agent Apache monitoring solution. You can download the code for this example at:

http://www-18.lotus.com/wps/portal/tm

This solution provides the capability of monitoring an Apache Web server using the Universal Agent. The Apache monitoring solution uses the File data provider to extract useful attributes about the health of your Apache Web server. This provides you with useful data about the performance characteristics of your Apache server including:

- Throughput of your Web server
- Number of Web hits per hour/day
- Errors including the client that initiated the request to the Web server

You can use the collected information to perform trending analysis on your Apache server. This solution will run on any platform version that is supported by the Universal Agent including Microsoft Windows, IBM AIX 5L, Sun Solaris, Linux, and HP/UX systems.

After the data is being collected by the Universal Agent, it is easy to modify the default queries and views to create workspaces that contain useful information. By counting the number of failed requests from a particular client IP address, a situation can be written to automatically detect for denial of service attacks or a user attempting to gain unauthorized access to your Web server.

The solution has been tested with Versions 1.x and 2.x of Apache. As long as the format of the access.log file has not changed, the solution should function on future versions of the product.

5.9.2 Installing the Apache Monitoring Universal Agent

The *readme* for the Apache Monitoring Universal Agent assumes that you are installing the agent on a Windows system. In order to install this agent on a Linux system, complete the following steps:

1. Ensure that the File data provider is enabled on your Universal Agent. By default, this is a setting in the <install directory> /config/um.config file. Set the default as follows:
   a. KUMA_STARTUP_DP='ASFS', where ASFS is the File data provider. Copy and point the metafile to the proper log location.

b. The metafile for any Universal Agent application must be placed in a specific directory on the host monitored by this agent.

c. In this case, copy the metafile (Apache.mdl is in that Apache.zip file that you downloaded from the OPAL site). Copy this file to the monitored host metafile directory. For example, <install directory>/li6243/um/metafiles, where <install directory> refers to the base directory where the Universal Agent was installed. In Linux, this directory defaults to /opt/IBM/ITM.

2. Set the metafile to use the access log file of the Apache server. With a text editor, for example vi, edit the metafile and look for a line starting with the metafile directive SOURCE, for example: //SOURCE file /usr/local/apache/logs/access_log tail.

3. Set the format of the access log. By default, the access log of the Apache server in Linux is set to Common Log Format (CLF). The metafile provided with this Apache Monitoring Universal Agent is set to work with a different and more descriptive log format called the Combined or NCSA extended format.

   a. To set the Apache server to use combined format, use a text editor, such as vi, and edit the configuration file <INSTALL_DIR>/conf/httpd.conf, where <INSTALL_DIR> is the installation directory for the Apache server. On our Red Hat system, this directory defaults to /usr/local/apache/.

   b. In this configuration file, look for the directive CustomLog, for example, CustomLog logs/access_log common, and change it to CustomLog logs/access_log combined.

   c. Save your changes and restart the server.

4. You are ready now to import the metafile into the Universal Agent and create an application. To do this, first make sure that the agent is up and running. Then change your current working directory to <install directory>/li6243/um and issue the following command, where Apache.mdl is the name of the metafile:

   ```
   ./bin/kumpcon import Apache.mdl
   ```

   The **kumpcon** command accepts the command, and after a brief moment, returns with a success message.

5. Verify the creation of the Apache application and its attribute groups. At this point, the Apache monitoring application should have been created and a new managed node named APACHE00 should appear in the physical view of the Tivoli Enterprise Portal. If you drill down this application, you should see its 20 attribute groups.

5.9.3 Apache Monitoring Universal Agent attribute groups

Table 5-4 lists the most commonly used attribute groups.

Table 5-4 Apache Monitoring Universal Agent attribute groups

Attribute group	Description
Exception_Detail	Reports Apache response codes greater than 400 in real time when they occur.
Workload_Detail	Summarizes the number of bytes received and sent by the Apache server.
HIST_HTTP_Stat	Summarizes HTTP responses by number.
HIST_Status_Stat	Summarizes HTTP responses other than 200.
HIST_Request_Stat	Summarizes HTTP requests.
HIST_Status_By_Request	Summarizes HTTP requests' statuses.
HIST_Referral_Stat	Summarizes HTTP referrals by status.
HIST_Transfer_By_Location	Summarizes HTTP transfers by location.
HIST_Transfer_By_Request	Summarizes HTTP transfers by request.
HIST_Workload_By_Day	Summarizes HTTP workload every 24 hours. The workload is expressed in terms of the number of bytes received and sent by the HTTP server.
HIST_Workload_By_Hour	Summarizes HTTP workload every hour. The workload is expressed in terms of the number of bytes received and sent by the HTTP server.
HIST_Referral_By_Location	Summarizes HTTP referrals by location.
HIST_Request_By_Location	Summarizes HTTP requests by location.
HIST_Browser_Stat	Summarizes browser IDs that have contacted the HTTP server.
HIST_Client_Platform_Stat	Summarizes the platforms (Windows, Linux, and so on) of the browsers that have contacted the HTTP server.

Because most of these attribute groups are summary attributes, they do not present data in Tivoli Enterprise Portal until the summarization cycle expires. This cycle for most of the attribute groups is one day by default (that is, 86400 seconds). This summarization cycle can be changed in the metafile for each attribute group by changing the value of the //SUMMARY control statement. Refer to the *IBM Tivoli Monitoring Universal Agent User's Guide*, SC32-9459, for more information about this control statement.

Figure 5-63 is an example of a Apache monitor workspace that summarizes the HTTP workload every hour. The workload is expressed in terms of the number of bytes received and sent by the Apache server.

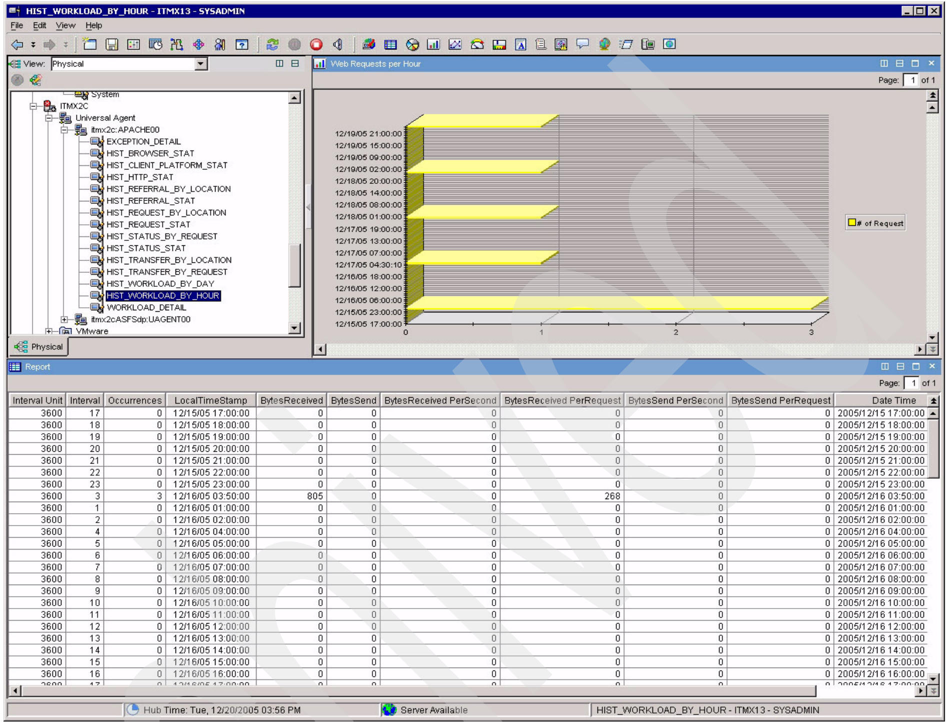


Figure 5-63 Apache workload by hour workspace

Figure 5-64 is an example of an Apache monitor workspace that shows response codes greater than 400 in real time when they occur.

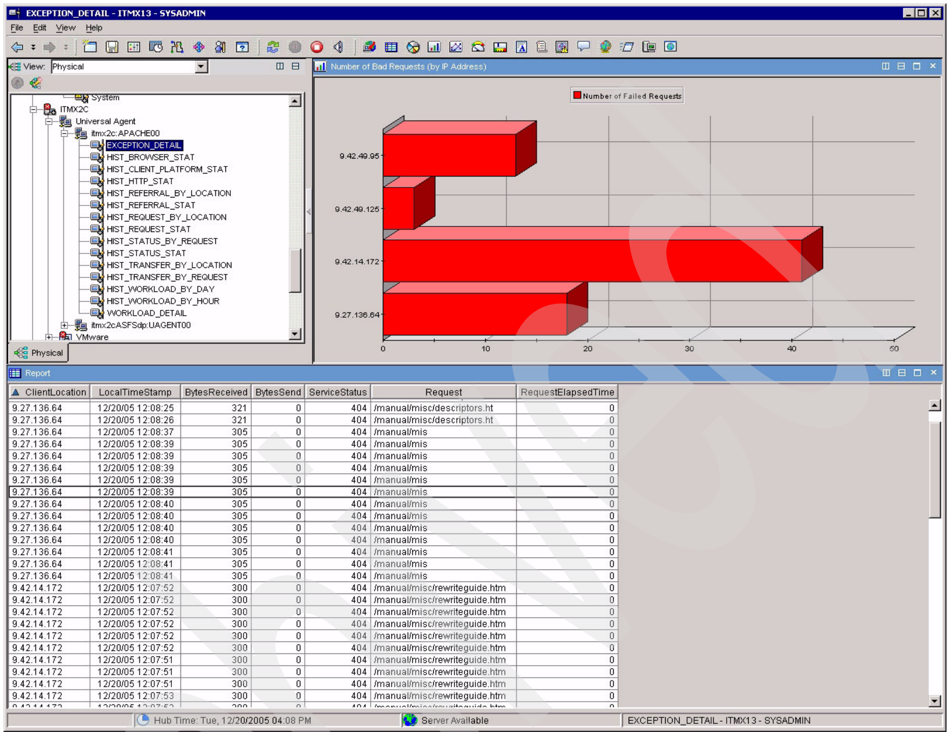


Figure 5-64 Apache exception details workspace

Figure 5-65 is an example of an Apache monitor workspace that summarizes HTTP transfers by location.

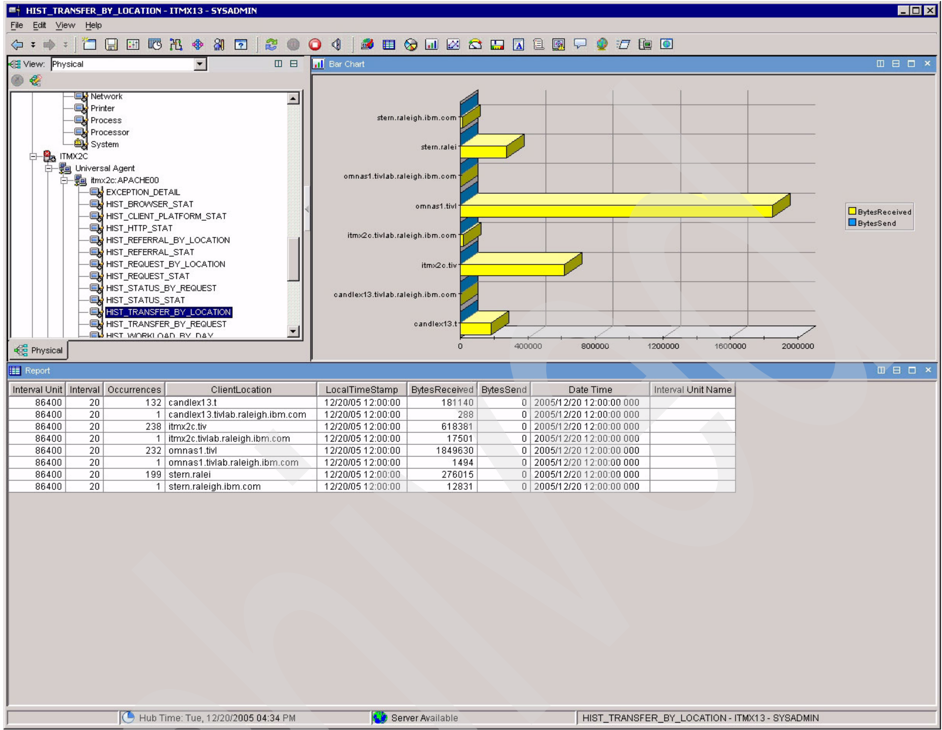


Figure 5-65 Apache HTTP transfers by location workspace

5.9.4 Custom situations for the Apache server

The Apache Monitoring Universal Agent does not come with any predefined situations. Therefore, all situations created for this solution will have to be custom. In this section, we use two new custom situations based on some best practice monitoring of an Apache server. Obviously, more situations can be created against any of the Apache attribute groups.

Table 5-5 presents the suggested best practice situations that we created.

Table 5-5 Best practice situations for the Apache server

Custom situation	Description
ITSO_Service_Status_Warning	This situation triggers if the number of HTTP error status codes (those more than the value of 400) have a high occurrence within the interval sample time (currently 5 minutes). If too many HTTP error status codes are received during this time, it is a signal that there might be something wrong with the HTTP elements requested, such a as a dead link or a non-existing image.
ITSO_Location_Traffic_Warning	This situation triggers if the amount of HTTP traffic coming from a single point or node is too high. If too many requests are received by the HTTP server from a single location, this might signal a security problem such as denial of service attack.

Figure 5-66 displays the ITSO_Service_Status_Warning custom-developed situation from the Situation Editor in the Tivoli Enterprise Portal.

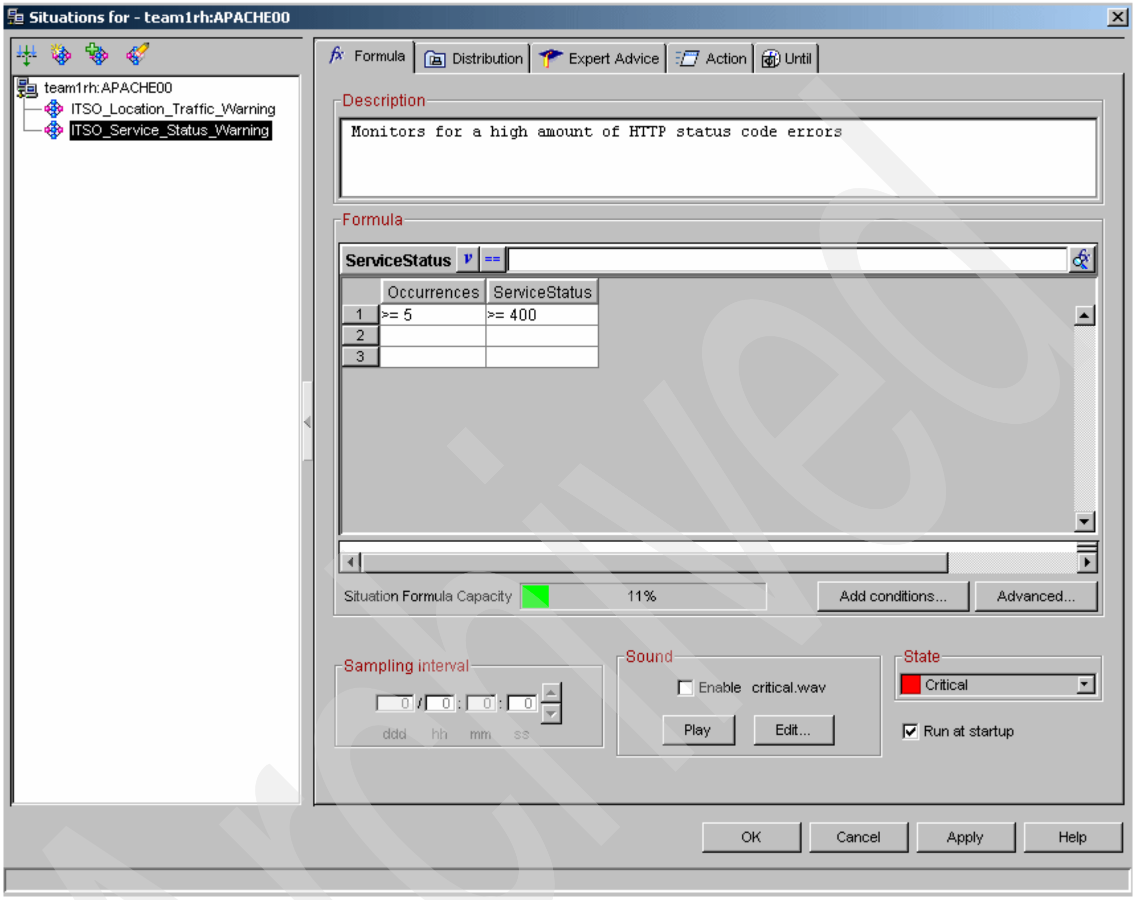


Figure 5-66   ITSO_Service_Status_Warning situation

Figure 5-67 shows the ITSO_Location_Traffic_Warning custom-developed situation from the Situation Editor in the Tivoli Enterprise Portal.

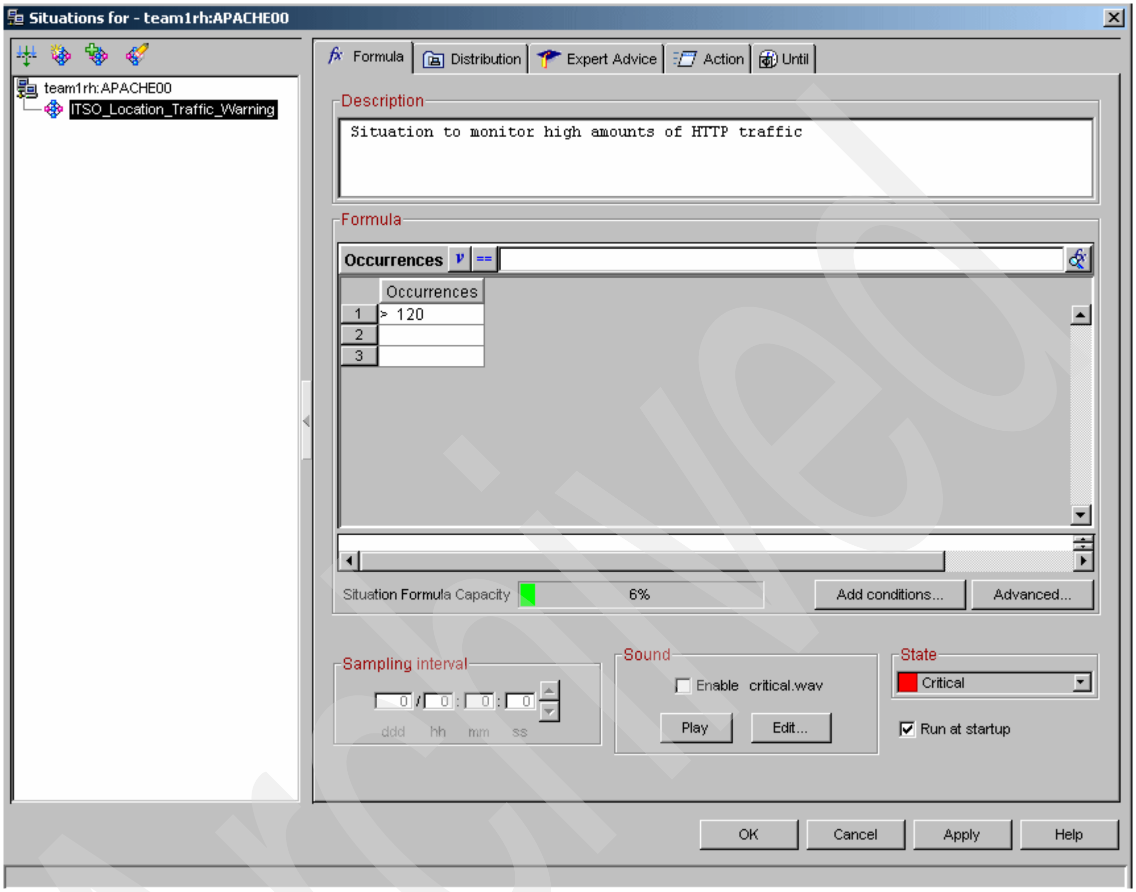


Figure 5-67 ITSO_Location_Traffic_Warning situation

5.10 Microsoft Internet Information Services scenarios

IBM Tivoli Monitoring Express does not have a unique agent for monitoring Microsoft Internet Information Services (IIS). However, it does supply generic attribute groups under the Windows OS agent that can be used to monitor IIS. The IBM Tivoli Monitoring Windows OS Agent is shipped with a set of predefined attributes that can be used to monitor IIS. These attributes can be used in custom situations to monitor the health of IIS. Custom situations can contain attributes that check for system conditions common to many businesses.

5.10.1 Best practice attribute groups to monitor IIS

The IBM Tivoli Monitoring Windows OS Agent contains more than 70 attribute groups that contains thousands of OS attributes that can be monitored on a Windows system. In this section, we look at six specific attribute groups with attributes that can be helpful in monitoring the health of IIS.

Table 5-6 lists the most commonly used attribute groups.

Table 5-6 Attribute groups for Microsoft Internet Information Services

Attribute group	Description
Active Server Pages	Use the Active Server Pages attributes to create situations to monitor Active Server Page requests, session information, and memory allocation. Active Server Pages is a single-instance attribute group.
HTTP Content Index	Use HTTP Content Index attributes to monitor queries made to an HTTP server, such as the number of active queries, the current requests queued, and the percentage of queries found in the query cache. HTTP Content Index is a single-instance attribute group.
IIS Statistics	Use IIS Statistics attributes to monitor memory usage and connection data. IIS Statistics is a single-instance attribute group.
Web Service	Use Web Service attributes to create situations to monitor traffic and connection activity for a Web server. Web Service is a multiple-instance attribute group. You cannot mix these attributes with those of any other multiple-instance group.
NT_Service Dependencies	Use NT_Service Dependencies attributes to obtain configuration information about all of the services or load order groups that must start before a given service installed on the Windows server. Services are background processes run by the operating system, regardless of the user logged on to the system. NT_Service Dependencies is a multiple-instance attribute group. You cannot mix these attributes with those of any other multiple-instance group.

5.10.2 Using custom situations to monitor IIS

This section identifies custom situations that are best suited for the needs of small and medium businesses. If necessary, you can change the conditions or values being monitored by a custom situation to those best suited to your client's business.

You can create custom situations using the Situation Editor in the Tivoli Enterprise Portal client. The left frame of the Situation Editor initially lists the situations associated with the Navigator item that you selected.

In this example, we use custom situations based on some of the default Windows OS agent attribute groups. With the Tivoli Enterprise Portal, you can view the data in real-time and historical workspaces on the Tivoli Enterprise Portal and manage with Tivoli Enterprise Portal monitoring situations and automation policies, the same as data from other Tivoli Enterprise Monitoring Agents.

Most of the attributes you need to monitor Microsoft IIS are included in the Windows OS agent. You can download the custom situation examples in this section from the ITSO Web site. For downloading instructions, refer to Appendix A, “Additional material” on page 393.

In addition, refer to the following Web site for additional custom situations:

http://sourceforge.net/projects/gulfsoft

In this section, we use three new custom situations based on some best practice monitoring of IIS. Obviously, you can create more situation for any of the Windows OS attribute groups. Table 5-7 lists best practice situations.

Table 5-7 List of best practice situations

Custom situation	Description
OpenESM_IIS_ASP	This situation monitors the Active Server Pages: Requests Executing. This counter measures the number of requests currently executing. This counter indicates whether the application is effectively executing one request at a time or not. If the requests executing is just 1, requests are being serialized for some unknown reason. A common source of the serialization is if you have turned ASP debugging on through Internet Services Manager.
OpenESM_IIS_Requests	This situation monitors the Web Service: CGI Requests/sec and ISAPI Extension Requests/Sec. These counters measure the rates at which your server is processing CGI and ISAPI application requests. If these values decrease while the load is increasing, you might want to have your application developers revisit their code.
OpenESM_IIS_Service	This situation monitors the NT Service attribute group to check if the IIS service is running.

Figure 5-68 displays the OpenESM_IIS_Requests custom-developed situation from the Situation Editor in the Tivoli Enterprise Portal.

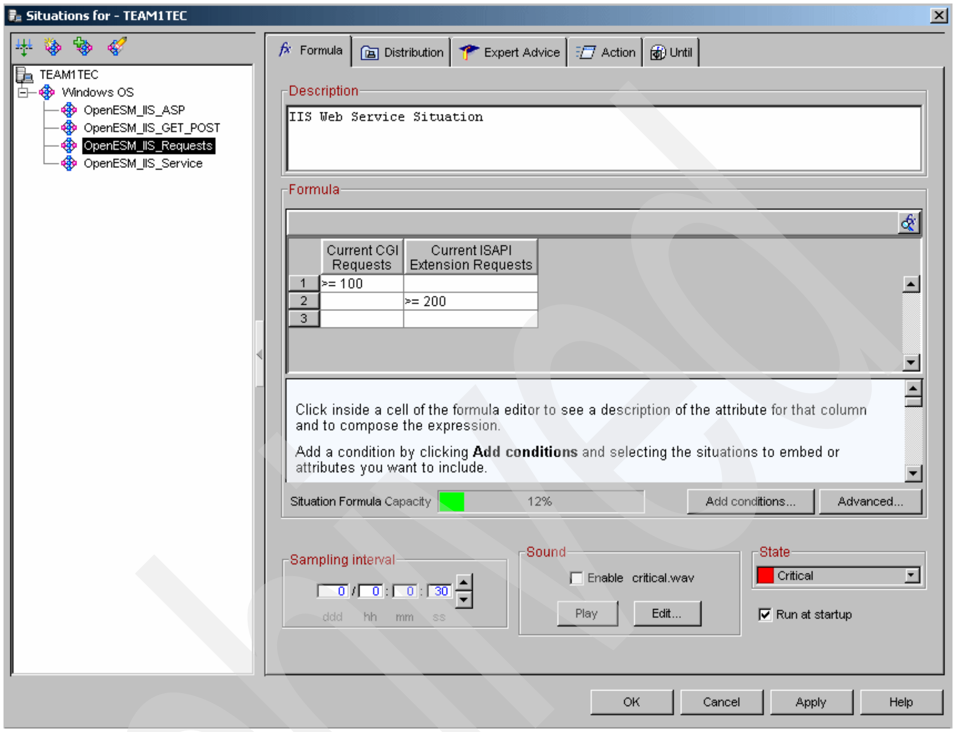


Figure 5-68 OpenESM_IIS_Requests situation

Figure 5-69 displays the OpenESM_IIS_Service custom situation from the Situation Editor in the Tivoli Enterprise Portal.

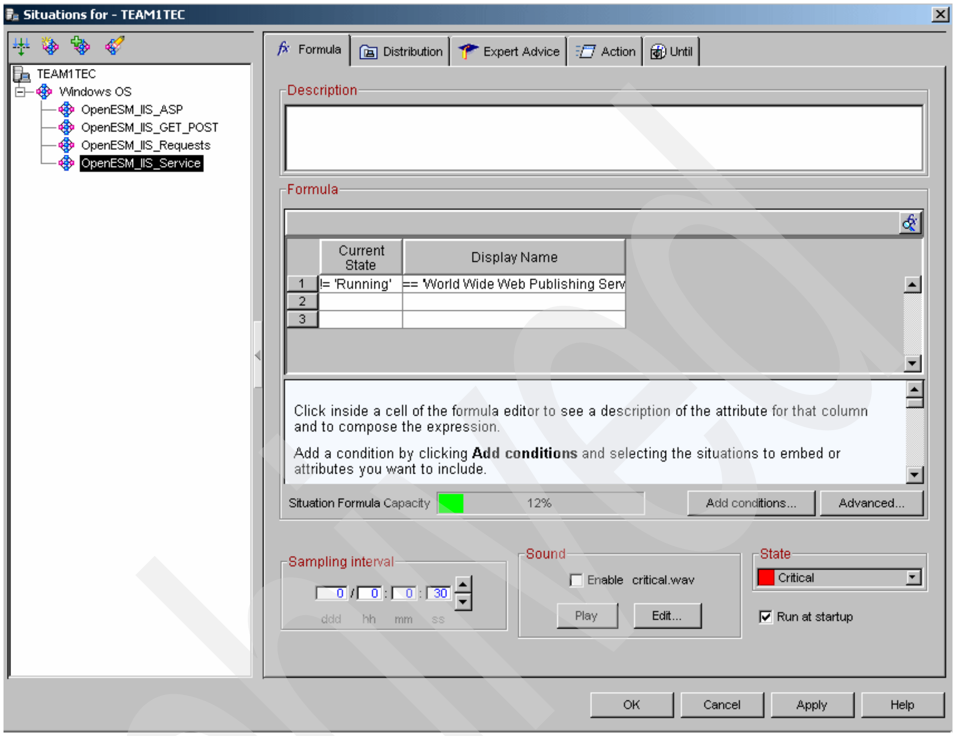


Figure 5-69 OpenESM_IIS_Service situation

Figure 5-70 shows an example of an Apache monitor workspace that shows response codes greater than 400 in real time when they occur.

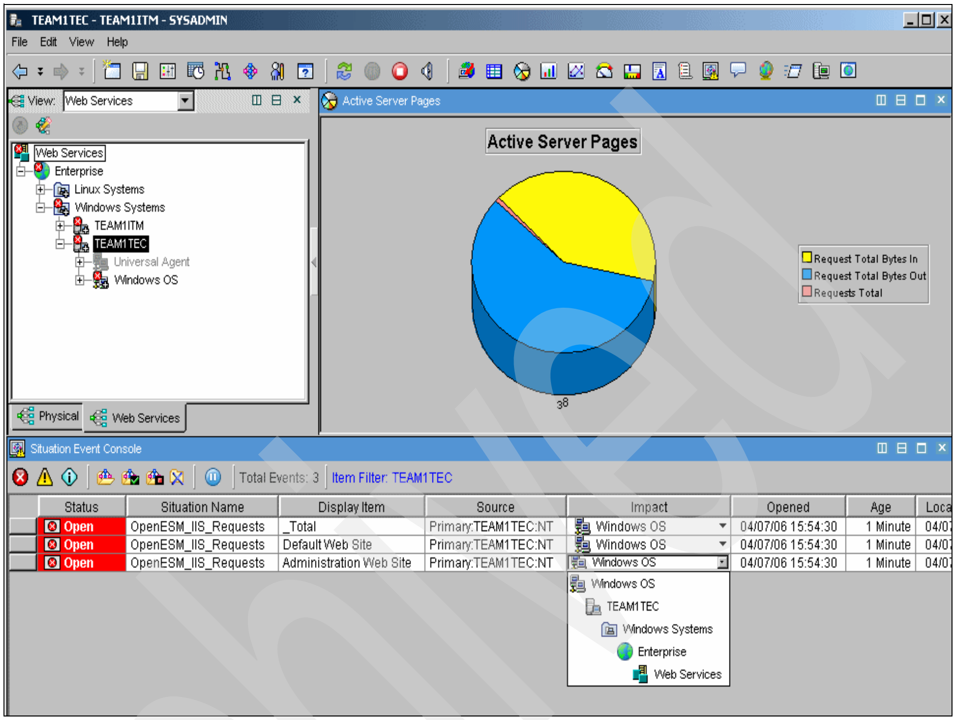


Figure 5-70   Active Server Pages workspace view

5.11 Microsoft SQL Server scenarios

The IBM Tivoli Monitoring Agent for Microsoft SQL Server is shipped with a set of predefined situations that you can use as-is or you can create new situations to meet your requirements. Predefined situations contain attributes that check for system conditions common to many businesses.

You can display predefined situations using the Situation Editor in the Tivoli Enterprise Portal client. The left frame of the Situation Editor initially lists the situations associated with the Navigator item that you selected.

This section identifies the predefined situations that are best suited for the needs of small and medium businesses. If necessary, you can change the conditions or values being monitored by a predefined situation to those best suited to your client's business. Table 5-8 on page 294 lists predefined situation for Microsoft SQL Server.

Table 5-8 Predefined situations for Microsoft SQL Server

Predefined situation	Description
MS_SQL_Cache_Hit_Ratio_Warn	This situation provides an indication of the cache hit ratio being less than 90% but greater than 70%. It triggers a warning alert if the ratio of data cache hits to total data requests exceeds the warning threshold.
MS_SQL_Cache_Hit_Ratio_Crit	This situation monitors if the database performance is being severely impaired. Default threshold is < 70%. It triggers a warning alert if the ratio of data cache hits to total data requests exceeds the warning threshold.
MS_SQL_Client_Cnt_Pct_Used_Crit	This situation triggers if the system is almost out of client licenses. Default threshold is > 90% of the licenses are currently in use. It issues an alert if the percentage of client licenses being used exceeds the critical threshold.
MS_SQL_DB_Error_Status	This situation monitors the database and triggers an alert if the database has a serious error and corrective action is needed.
MS_SQL_DB_Freespace_Critical	This situation triggers an alert if the database is nearly reaching its maximum size. Default threshold is freespace <= 10%. It declares a critical condition if the percentage of free space on the database is less than or equal to 10.
MS_SQL_DB_Suspect_Crit	This situation monitors the database and triggers a critical alert if the database is in an inconsistent or "suspect" state because it cannot be restored.
MS_SQL_Block_Critical	This situation triggers on the number of blocked processes. It declares a critical condition if the number of processes in conflict is greater than 60.
MS_SQL_IO_Disk_Errors_Crit	This situation triggers a critical alert if the number of SQL Server read/write disk errors exceeds the critical threshold.
MS_SQL_Status_Critical	This situation monitors the SQL Server and declares a critical condition if the SQL Server status is inactive.
MS_SQL_Logon_Pct_Critical	This situation triggers an alert if the number of logons is getting close to the maximum. Default threshold is >= 90%.
MS_SQL_PCT_MAX_Locks_Critical	This situation issues a critical alert when the percentage used in secondary log exceeds the critical threshold.
MS_SQL_Repl_Latency_Crit	If replication is used, MS_SQL_Repl_Latency_Crit provides an indication of replication latency reaching a critical value.

5.12 IBM Tivoli Universal Agent scenarios

IBM Tivoli Universal Agent (UA) is a generic agent used in conjunction with other Tivoli Enterprise Monitoring Agents to collect data and monitor systems and applications in the network. In this section, we discuss the following IBM Tivoli Universal Agent topics:

- What is Tivoli Universal Agent?
- IBM Tivoli Universal Agent architecture
- Universal Agent deployment scenarios

5.12.1 What is Tivoli Universal Agent?

IBM Tivoli Universal Agent is a generic agent used in conjunction with other Tivoli Enterprise Monitoring Agents to collect data and monitor systems and applications in the network. In turn, this data can be used and visualized in the Tivoli Enterprise Portal. You can use all standard Tivoli Enterprise Portal 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 hardcoded attributes, in other words, predefined data. Therefore, they cannot be enhanced by the field personnel to “see” more than for which they are developed. However, the Universal Agent is a full-featured intelligent remote agent (IRA) 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, you can generate better alerts to give more information about the health of an installation. The data collected by the Universal Agent can be used for specific monitoring actions through the Tivoli Enterprise Portal situations.

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 easily be made to support new features in an application.

- Monitors data not supported by the 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.

5.12.2 IBM Tivoli Universal Agent architecture

Figure 5-71 shows the high-level architecture and data flow for the Universal Agent.

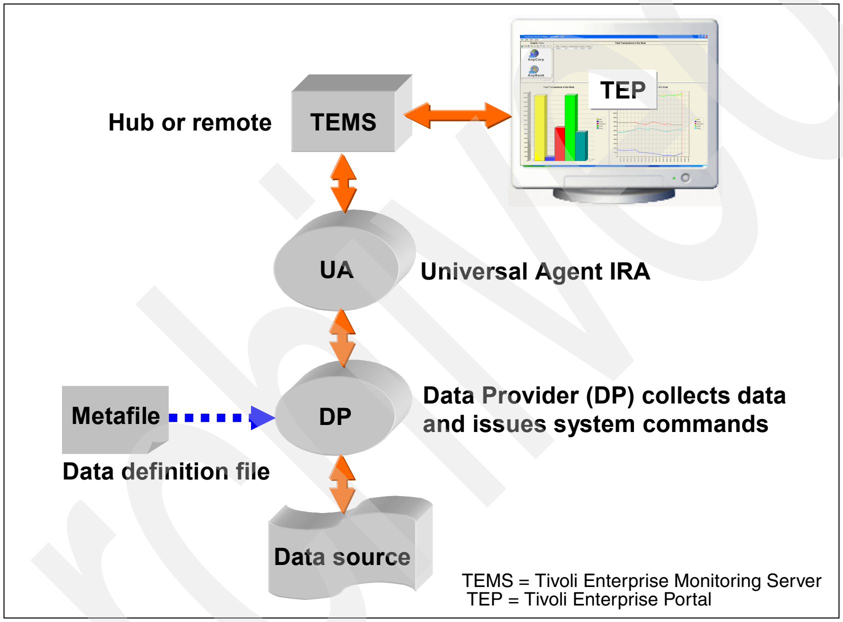


Figure 5-71 High-level architecture and data flow for Universal Agent

The data source for the Universal Agent is something that the installation provides. It can 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 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 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 Universal Agent 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.

Universal Agent data provider

Data is collected from the monitoring environment and passed to IBM Tivoli Universal Agent through structures called *data providers*. These data providers work as IBM Tivoli Universal Agent threads, using applications called metafiles 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.

You can choose from the following nine data provider categories depending on your monitoring requirements:

- API Server
- API, Socket, File, Script (ASFS)
- File
- HTTP
- ODBC
- Post
- Script
- SNMP
- Socket

Table 5-9 lists the data providers available with IBM Tivoli Universal Agent.

Table 5-9 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 Universal Agent.
File	Monitors sequential files, such as system or message logs. Provides the most direct and simplest method of collecting data.
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. Available only 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 Universal Agent API support is available.

The right choice of a data provider depends on the type of data you 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 is a Socket data provider.

Table 5-10 lists data sources and related data providers.

Table 5-10 Data source and preferred data providers

Data source	Preferred data provider
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
Any combination of the following items: ► 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 when you install the Universal Agent. It consolidates four types of data providers, that is, API, Socket, File, and Script into one package, which is started as a single thread to save resource usage.

The Universal Agent has the ability to run several instances of the same data provider on the same monitored host. The reasons for this is because it can:

- Run test and production versions of the Universal Agent on the same host.
- Balance the data load of a Universal Agent that is overloaded.
- Connect to several Universal Agents on different Tivoli Enterprise Monitoring Servers.

A 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 5-11 lists the typical Universal Agent ports.

Table 5-11 Typical ports used by the 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
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. You can change this port to access a secondary Universal Agent using the KUMP_DPCONSOLE_PORT variable to specify the alternate port number.

Collecting and monitoring Universal Agent metafiles

With applications called metafiles, you can 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 Serve, similar to the data collected by specific Tivoli Monitoring Agents.

Make a metafile application consisting 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

Table 5-12 shows the metafile control statements, if present.

Table 5-12 Metafile control statements

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. Lists the individual attribute definition statements.

Example 5-5 shows a sample of a metafile that maps log files. Each log file is identified as a separate managed system. TAIL tells the Universal Agent that you are going to read records from the end of the file as they are written.

Example 5-5 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 you 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.

A metafile has both a version and modification number. When it is imported for the first time in the IBM Tivoli Universal Agent, it is assigned a version number of 0 and a modification number of 0. When changes are made in the metafile and it is refreshed on the Universal Agent, the version or modification number is incremented by one, depending on the type of the modification.

Changes that do not affect the version or modification number of the metafile include:

- Changing TTL value
- Changing to the SOURCE statement
- Changing the data type from P, S, or K to any of P, S, or K
- Changing Delimiter specified in the ATTRIBUTE statement
- Changing to the RECORDSET statement
- Changing to the CONFIRM statement
- Changing to attribute FILTER parameters
- Changing to the SQL statement

The following minor changes affect the modification number:

- 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 major changes increment the version number:

- 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
- 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
- Adding a new attribute anywhere other than at the end of a list of existing attributes

Creating metafiles will be much clearer when you read through the scenarios in 5.12.3, "Universal Agent deployment scenarios" on page 306.

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 Tivoli Enterprise Portal.

Tivoli Enterprise Portal objects are called managed systems, and the name of each managed system identifies the collected data source, the application that has been monitored, and the metafile version. Tivoli Enterprise Portal can configure workspaces to visualize the data collected by the Universal Agent. Each attribute group defined in a metafile has its own workspace. It can also be customized to show only wanted data.

The attribute groups DPLOG and ACTION from each data provider are used for self-monitoring, more specifically data providers. The DPLOG attribute shown in Figure 5-72 displays the status from a data provider, and the ACTION attribute provides information about the execution of a situation.

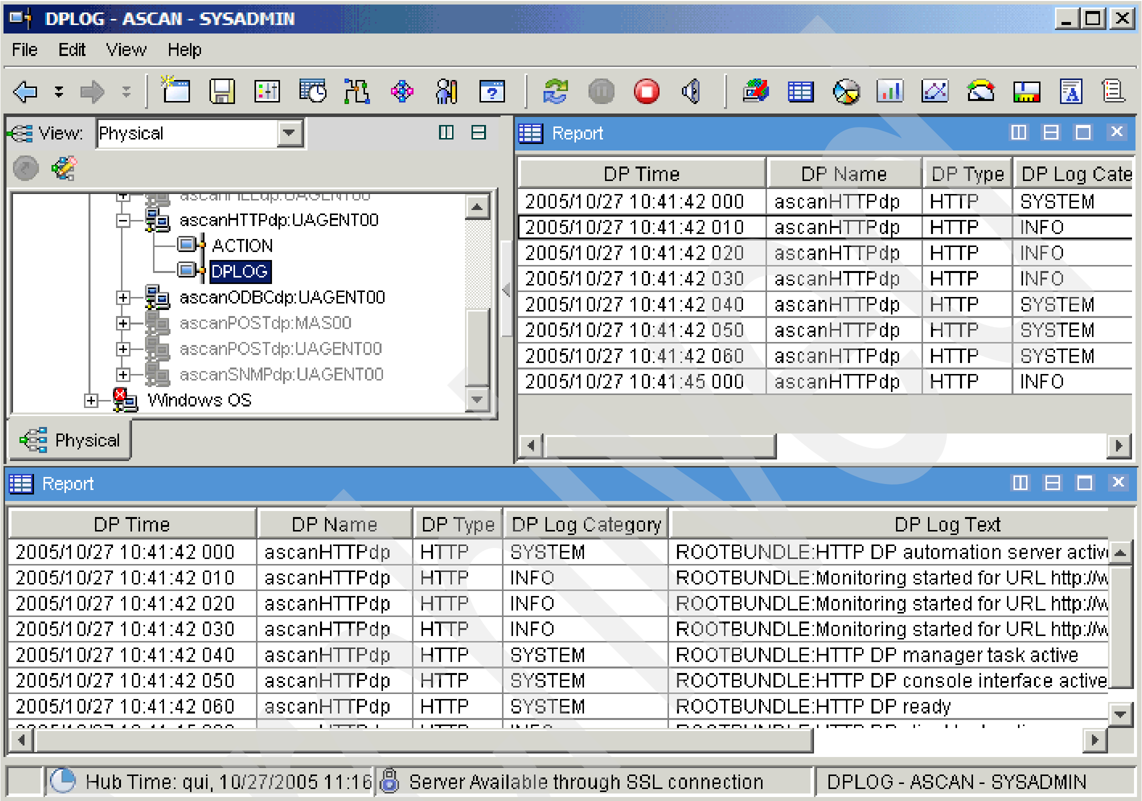


Figure 5-72 Attribute group DPLOG in Tivoli Enterprise Portal

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 Tivoli Enterprise Portal, you can visualize the data collected and stored by the Universal Agent with the historical data collection functionality.

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, whether 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-monitor data providers.

When using Universal Agent is a good choice

The Universal Agent is a good choice when, for example, systems and applications cannot be monitored by existing monitoring solutions, when you 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.

Consider the following points before deploying the Universal Agent:

- Choose the right data provider for your application monitoring. For example, 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 fire by some systems monitoring conditions.

Some examples of real-world Universal Agent usage include:

- Monitoring WebSphere MQ client channels
- Monitoring DEC OpenVMS
- Integrating Cabletron Spectrum

- Monitoring remote RF devices
- Monitoring point of sale (POS) devices
- Monitoring proprietary applications

5.12.3 Universal Agent deployment scenarios

To deploying a Universal Agent (UA) solution, perform the following steps:

1. Collect all the required information about the solution.
2. Select the data provider and start the UA with the selected data providers.
3. Create the metafiles describing the Universal Agent application.
4. Load the metafile and send the data.
5. Use standard IBM Tivoli Monitoring Express features to finalize the solution.

The first step is especially important. Answering the following questions might help you determine how to collect all the 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 gathering this information, determine the correct data provider type to use. This decision will be based on the information you collected.

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

HTTP data provider deployment scenario

The IBM Tivoli Monitoring HTTP data provider provides the ability to monitor URLs under some conditions such as availability and response time. The HTTP data provider, which is different from other data providers, does not use a definition file (metafile).

Starting the HTTP data provider

Start the HTTP data provider the same way you 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, use the GUI.

> **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 in the following conditions:
>
> - Outside a firewall
> - On a special machine with limited resources
> - To monitor a file on a remote system
>
> To start the HTTP data provider as a separate process, invoke the following program:
>
> ```
> KUMPHTTP
> ```

To start the HTTP data provider from the GUI, perform the following tasks:

1. In the Manage Tivoli Enterprise Monitoring Services window, right-click **Universal Agent** and select **Reconfigure** (Figure 5-73).

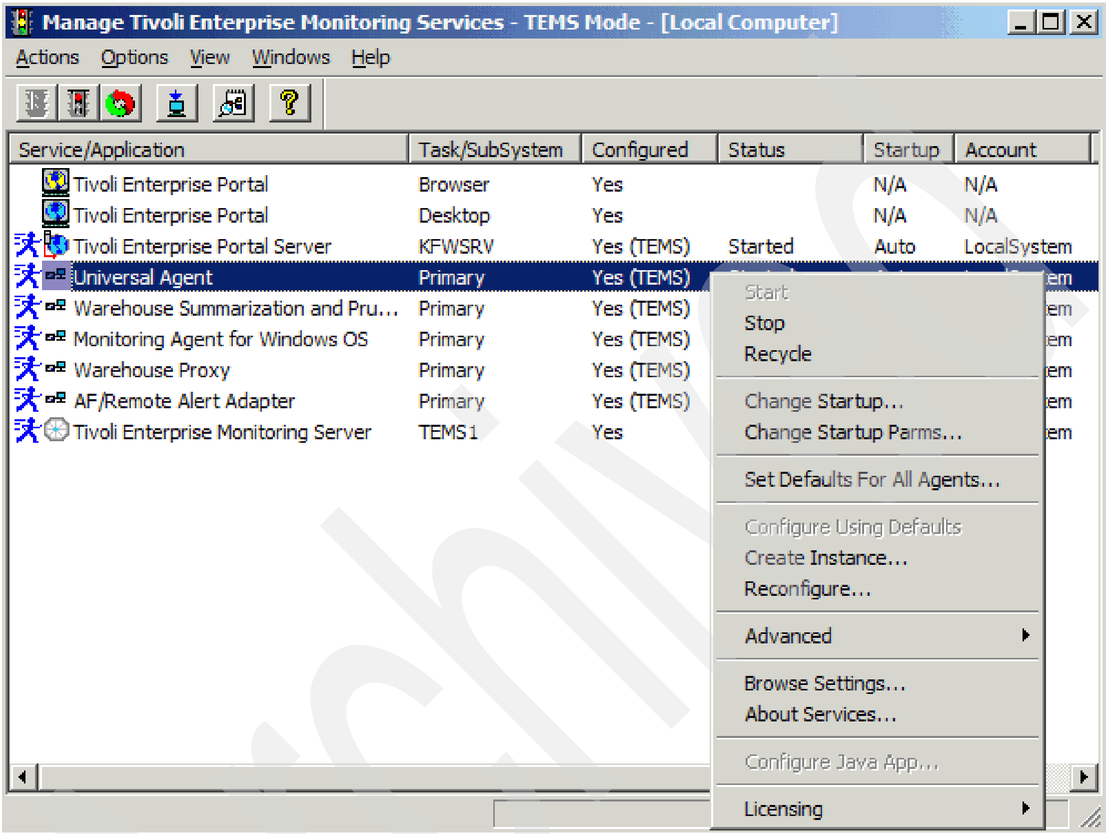


Figure 5-73 Changing the startup parameters of data providers

2. In the next two windows, click **OK**.

3. When prompted to update the KUMENV file to configure the Universal Agent, click **Yes** (Figure 5-74).

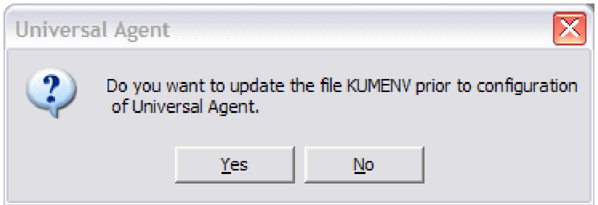


Figure 5-74 Updating the KUMENV file

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 the Universal Agent (Figure 5-75). It will be stopped but not configured yet.

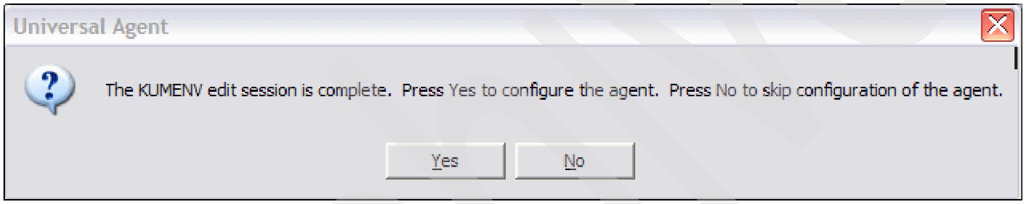


Figure 5-75 Configuring the Universal Agent

Configuring URL monitoring

Now, configure some URLs to monitor using the HTTP data provider. To monitor a URL, configure the KUMPULRS file located in the work directory and define the URLs that you want to monitor, as shown in Example 5-6. The KUMPURLS is a definition file used to monitor URLs. If this file does not exist, you can only monitor URLs using situations or Take Action. However, it is not necessary to use the HTTP prefix.

Example 5-6 Monitoring URLs

```
* 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 the **Take Action** → **URL Add** selection. This is the procedure described in the “HTTP Data Provder” section of the *IBM Tivoli Universal Agent V6.1.0 User's Guide,* SC32-9459. It has the added benefit of not having to recycle the Universal Agent for the new monitoring to take effect.

Start the Universal Agent. After a few minutes, the monitoring begins. Figure 5-76 shows URL monitoring with a bar chart.

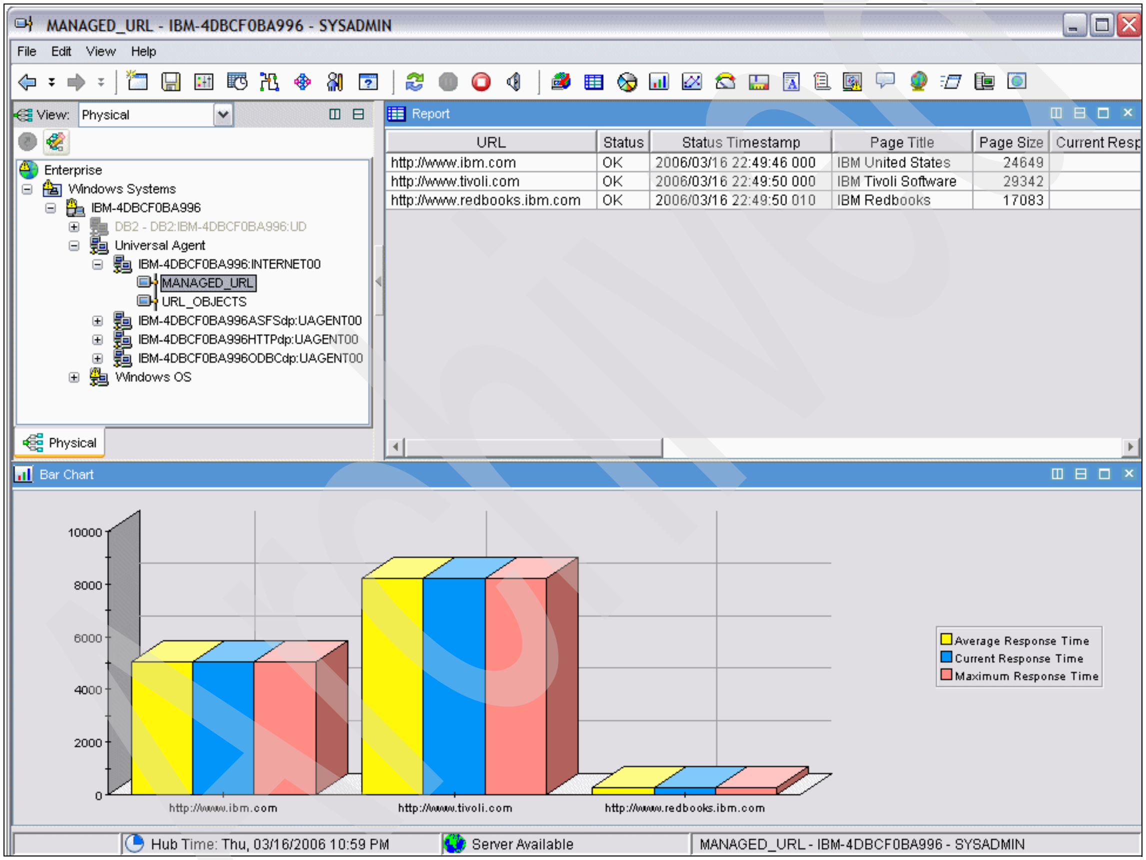


Figure 5-76 URL monitoring in Tivoli Enterprise Portal

Managed systems of the HTTP data provider

The HTTP data provider has the following managed systems, one for the Internet and one for the HTTP data provider:

- For the Internet:

  `host-name:INTERNET00`

- For the HTTP data provider:

  `host-nameHTTPdp:UAGENT00`

URL attributes of the HTTP data provider

The attributes shown in Table 5-13 are available in IBM Tivoli Monitoring situations' URL monitoring and are displayed in the Managed URLs table. Table 5-13 displays the URL attributes.

Table 5-13 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. 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 5-14 lists the attributes for Internet table URL objects, used to monitor the availability and response time for embedded objects in the Web site.

Table 5-14 URL objects

Attribute name	Type	Size	Description
Object Name	UTF-8	512	The name of the page object within the target URL.
Object Size	Integer	Long	The size of the page object within the target URL.
URL	UTF-8	512	The target managed URL. Use the format `http://`.

Historical data configuration

Historical data is configured to store data from URL monitoring the same way other historical data is configured to store data to others agents.

To configure historical data, perform the following tasks:

1. In the main window, click **History Configuration** (Figure 5-77).

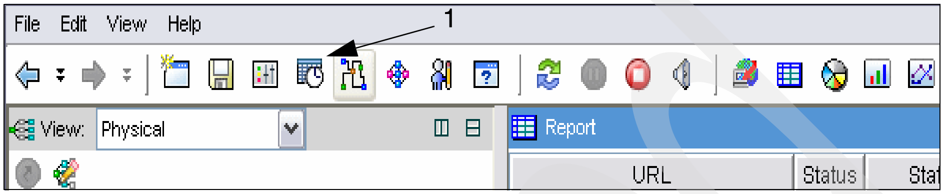


Figure 5-77 Selecting the Configuration icon

2. In the History Collection Configuration window, select **INTERNET** as the product.

3. To collect data about URL response time, you only have to configure the MANAGED_URL group, as indicated by the arrow in Figure 5-78.

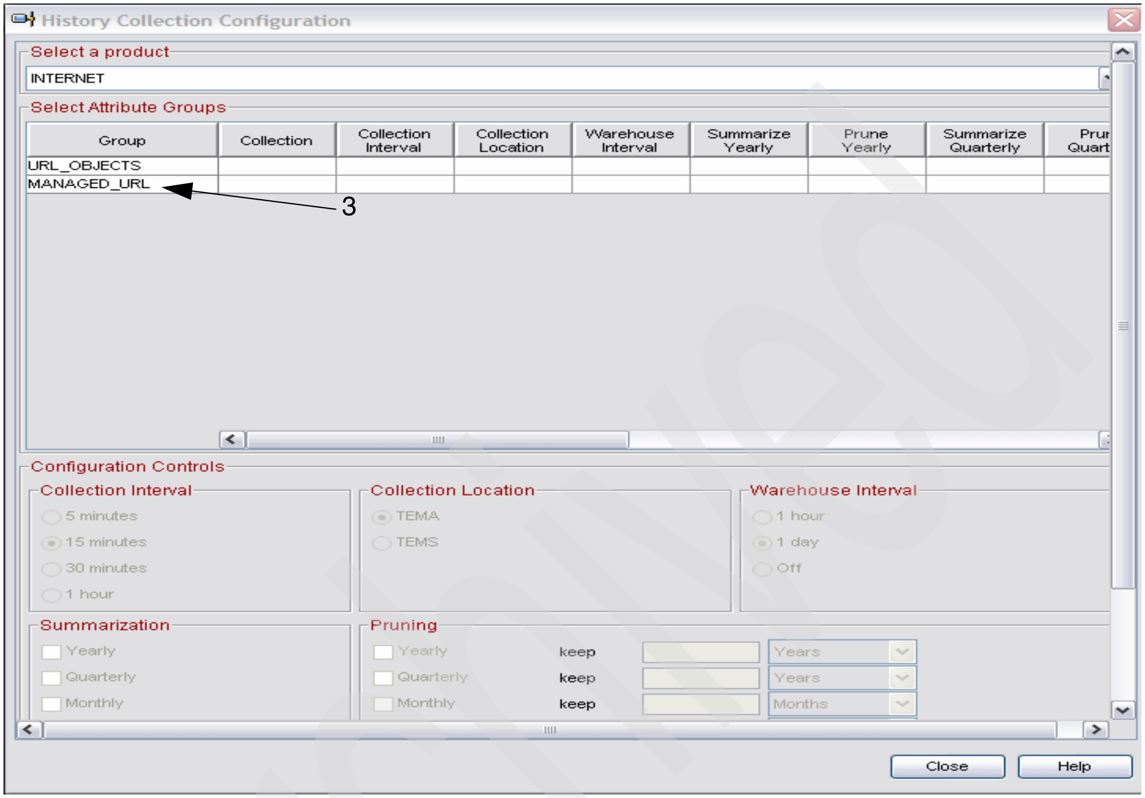


Figure 5-78 History Collection Configuration

4. In the next window (Figure 5-79), specify the following values:

   – Collection Interval
   – Collection Location
   – Warehouse Interval

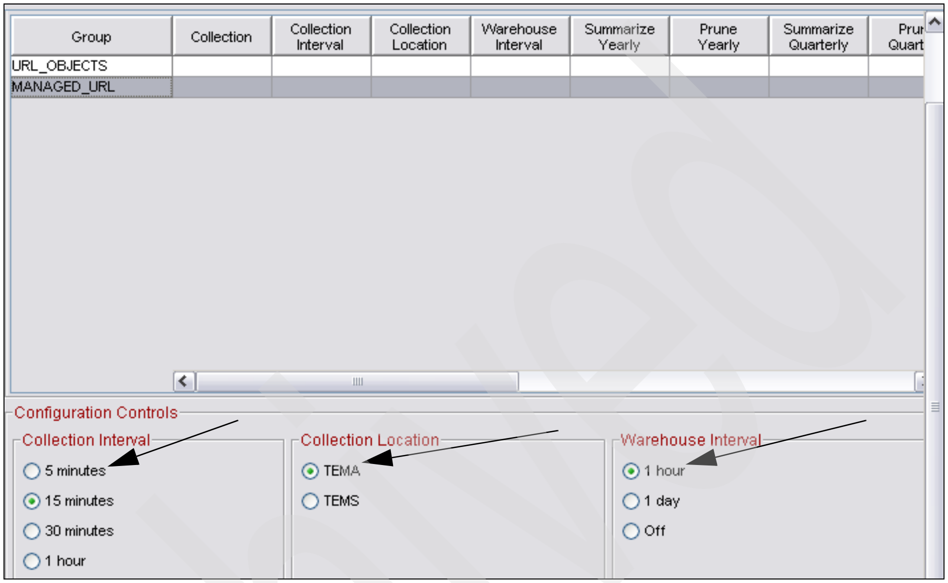


Figure 5-79 Historical data collection: Intervals and location

5. Click **Configure Groups** (Figure 5-80).

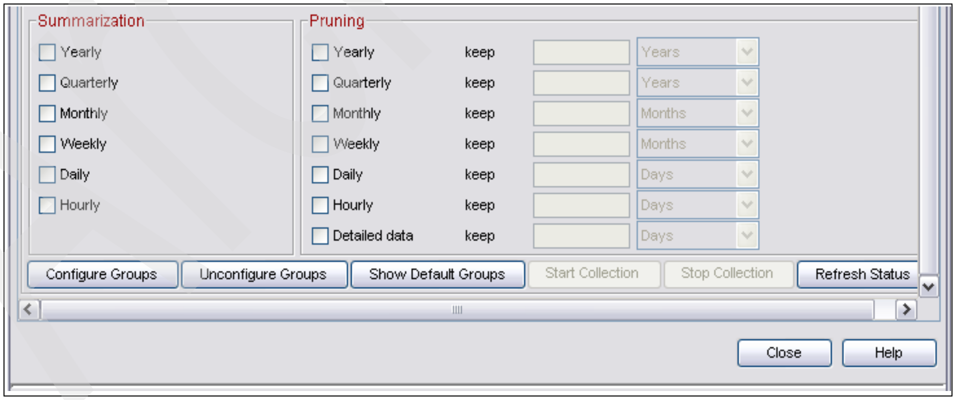


Figure 5-80 Historical data collection: Configure Groups

6. Highlight the **Group Managed_URL** at the top of the window and click **Start Collection** (Figure 5-81).

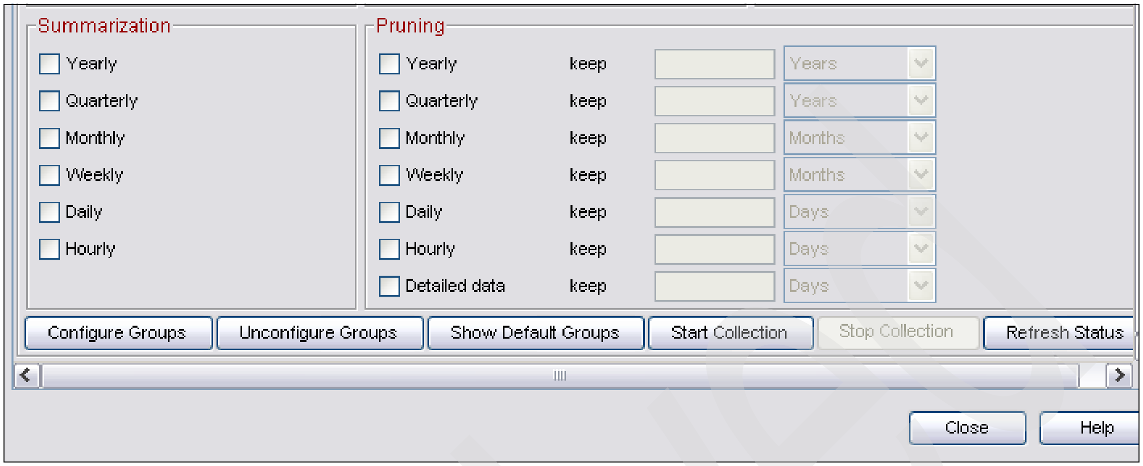


Figure 5-81 Historical data collection: Start Collection

Configuring situations

It is possible to create situations to be triggered for some conditions, for example, if a URL is not available. To do this, perform the following tasks:

1. Right-click **MANAGED_URL** in the physical tree and select **Situations** (Figure 5-82).

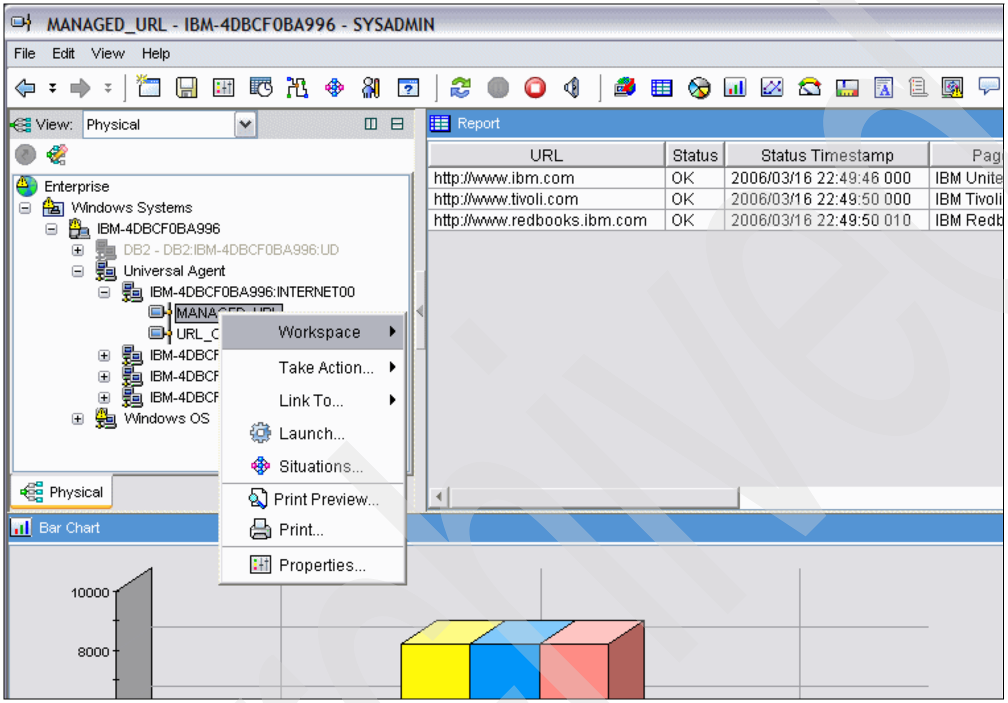


Figure 5-82 Situation configurations

2. In the Situations for - MANAGED_URL window, right-click **MANAGED_URL** (Figure 5-83) and select **Create New**.

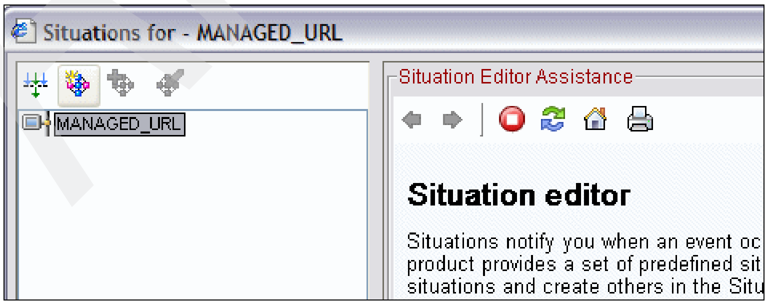


Figure 5-83 Create a new situation for HTTP data provider

3. Type a name and description in the Create Situation window.
4. In the Select condition window, select **MANAGED_URL** from Attribute Group list, and from Attribute Item group, select the items you want to monitor, that is, **Average Response Time**, **Status**, and **URL** (Figure 5-84).

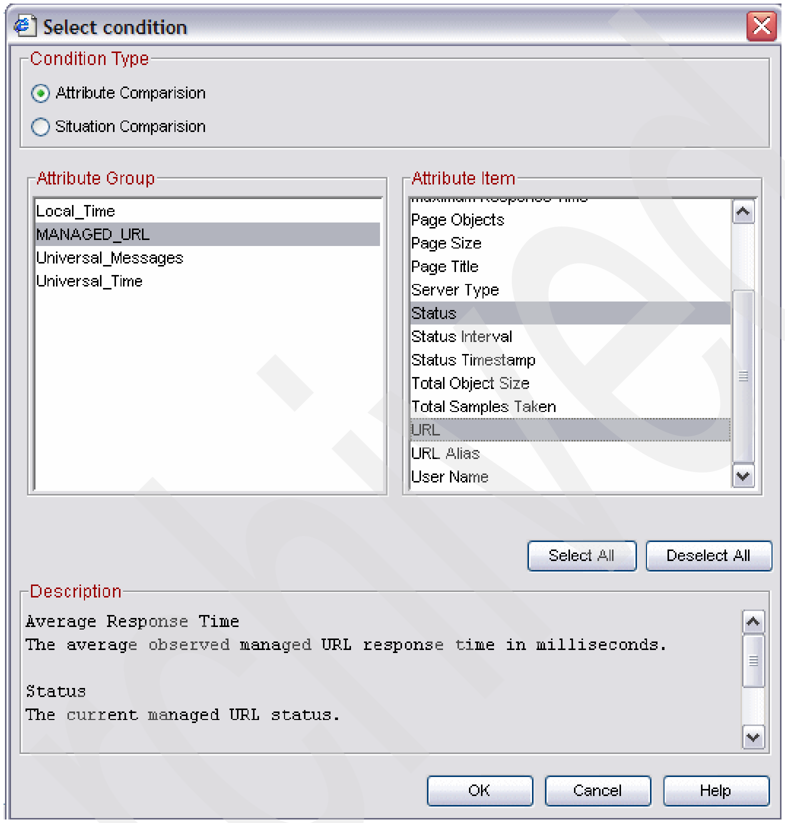


Figure 5-84 Attribute items selection window

5. The last step is to configure some conditions for the situation be triggered. In this example, for the three 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 five minutes, as shown in Figure 5-85.

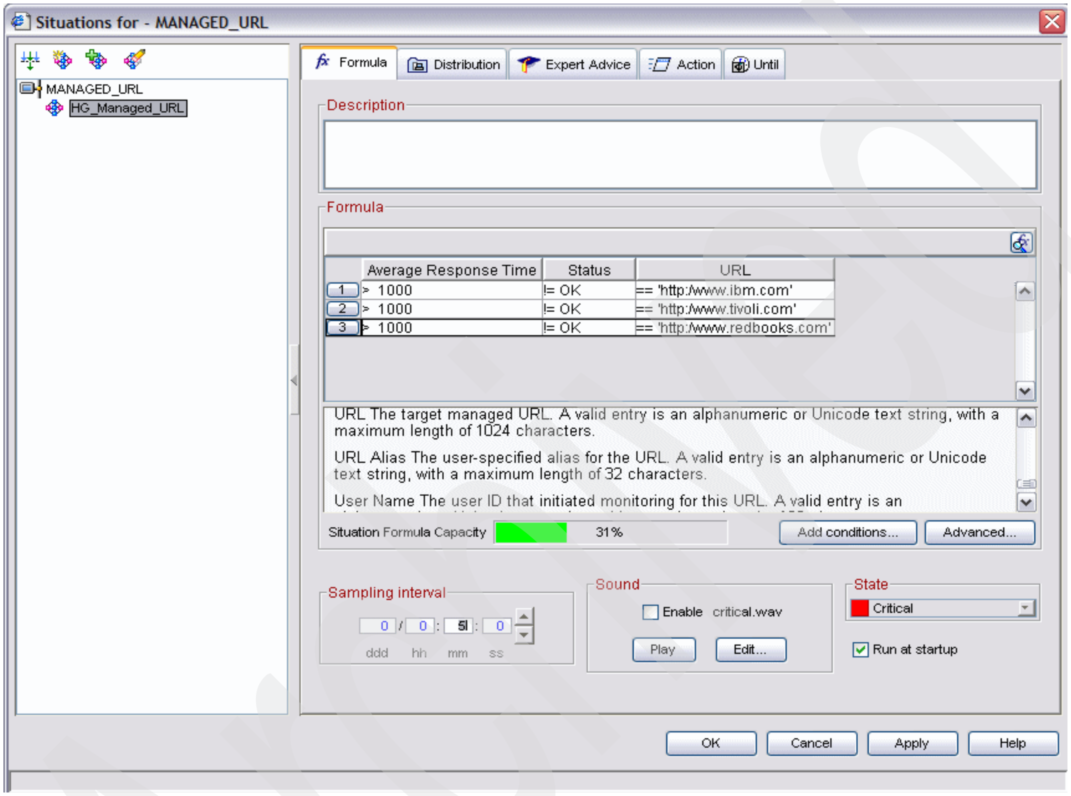


Figure 5-85 Situation parameters

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. It can also monitor networked files through NFS. It is the simplest way to monitor data using the Universal Agent.

Start a File data provider using the same method used to start other data providers, for example, configuring the KUMA_STARTUP_DP parameter in the KUMENV file:

```
KUMA_STARTUP_DP=FILE
```

> **Note:** You can also start a File data provider as a separate process using a command-line command to invoke the program:
>
> `KUMPFILE`

The File data provider must reside in the same host where you 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 the authority to open and read the file in the remote system.

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 five. The frequency for polled and sample data can be controlled using the KUMP_DP_SAMPLE_FACTOR environment variable.

The File data provider supports multiple record inputs when multiple physical file records comprise one logical record. For example, the data for two attributes can reside in one file record, and the data for a third attribute in another file record. This is when you should use the RECORDSET control statement in your metafile.

Applications can name output files based on several criteria such as day, week, or month. In our example, we specified a monitoring file name pattern in the //SOURCE statement as:

```
//SOURCE FILE file-name-pattern-spec
```

The File data provider inspects all the 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 and 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 5-15 lists file name pattern specification.

Table 5-15 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.

Consider the following pointers for 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 the 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, 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 5-16 lists the use of capital letters.

Table 5-16 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. It switches to a new file when a new file matches the defined pattern. This occurs under the following conditions:

- The File data provider first starts up.
- The existing file contents have changed.
- The check interval has expired.
- The currently managed file no longer exists due to possible renaming or deleting.

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 the stopping and starting of a NetView server.

Example 5-7 shows the metafile used in the File data provider deployment scenario to monitor the NetView log. The components of the metafile consist of:

1. The first line is the application name.
2. The second line is the name of the source
3. The data source file location and the TAIL tell the Universal Agent to look for any additions to the file as they are happening.
4. Attributes tell the Universal Agent how attributes are separated in the file, in this case, by space, because nothing follows the attribute command.
5. The next two lines define the eight characters for the date and time and define them to be of time Display (D).
6. In the log file, you can see that the source of the message is enclosed in square brackets. In this example, because we wanted only the source name without the brackets, we specified in the DLM parameter that the attribute is delimited by [ ].
7. 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.

Example 5-7 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
```

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 on the name. You can save the file with the name that you want.

Universal Agent has a validation program to check the metafile. 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, use the command in the first line of Example 5-8.

Example 5-8 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 5-9 shows a validation output file.

Example 5-9 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, specify to the Universal Agent what to monitor. You can do it in any of the following ways:

- Configure the KUMPCNFG file, located in the C:\IBM\ITM\TMAITM6\work directory, with the name of the metafile application and restart the Universal Agent service in the Manage Tivoli Enterprise Monitoring Services. For our example, nvlog, we use:

  ```
  * Universal Agent Configuration file
  nvlog
  ```

- Activate the metafile with console commands using IMPORT and REFRESH.

  If you use this method, you do not have to restart the Universal Agent. Example 5-10 shows the import of a metafile using the command line in Windows systems.

Example 5-10 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
```

- Activate metafile with Take Action commands in Tivoli Enterprise Portal.

Figure 5-86 shows the source NVLOG of the metafile after the import of the metafile application.

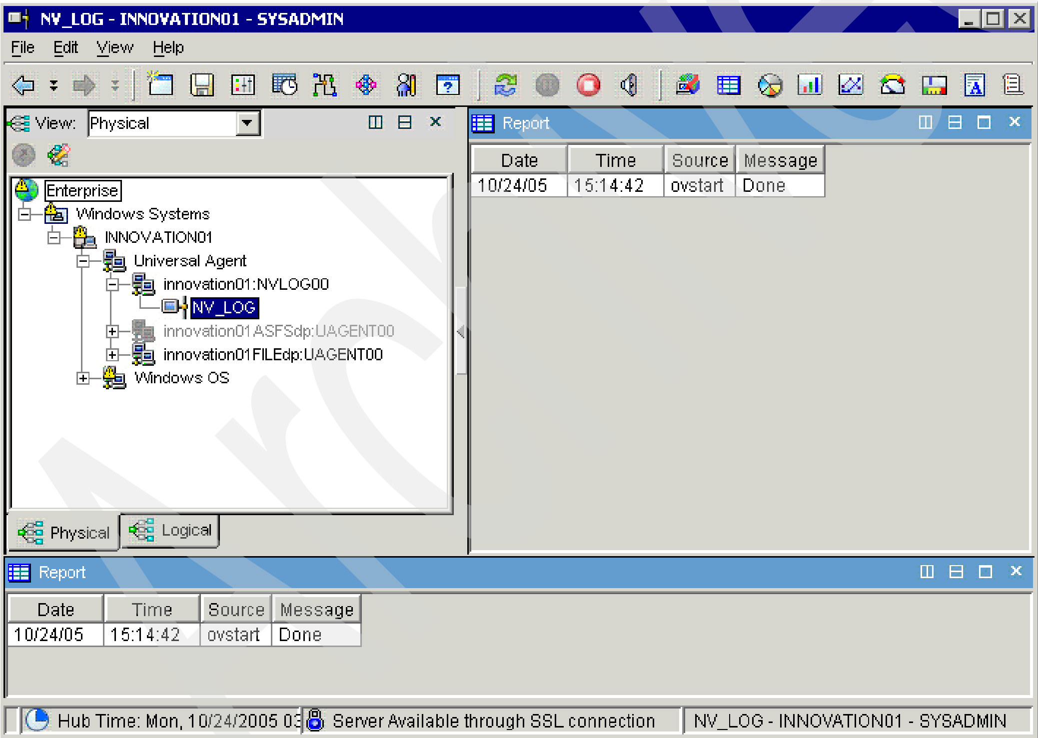


Figure 5-86 NV_LOG metafile source

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.

The OBDC data provider runs as a separate data provider and is available only on Windows operating systems. It runs on one machine and can simultaneously collect data from multiple remote databases.

The ODBC data provider is started the same way as other data providers, with the KUMA_STARTUP_DP parameter in the KUMENV file. For example:

```
KUMA_STARTUP_DP=ODBC
```

> **Note:** You can start the ODBC data provider as a separate process by invoking:
>
> `KUMPODBC`

Data can be collected as interval-driven or demand-driven data. The default, demand-driven data, signifies that data is collected only if a situation or report request is issued.

To create an ODBC metafile, use the **`generate`** command, instead of creating the metafile manually. This command saves you time and effort by creating an ODBC metafile automatically. With this command, you also limit which tables are generated, that is, user, system, view, or any combination.

However, this command does not support metafile generation for stored procedures. It is available only on Windows operating systems.

The correct syntax of the **`generate`** command is:

KUMPCON GENERATE *dataSourceName* user=*userid* pswd=*password*

Table 5-17 describes the parameters of the **`generate`** command.

Table 5-17 Generate command parameters

Parameter	Description
`dataSourceName`	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

To demonstrate how to use the ODBC data provider, we use a simple scenario, where we want to know all the users who are configured and logged on to the Tivoli Enterprise Portal Server database.

Generate the metafile used by the ODBC data provider using the **generate** command. Example 5-11 shows how to create a metafile using the **generate** command to monitor the TEPS2 database with ODBC data provider. Perform the following steps:

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 be included 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.
5. The metafile is created, as shown in Example 5-11.

Example 5-11 Metafile generation for the 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
```

6. Edit the metafile and include **select** statements only for KFWLOGIN and KFWUSER tables. Example 5-12 shows the metafile after these changes.

Example 5-12 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=itso05
//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
```

7. Before importing the metafile, it is necessary to validate it. Enter the following command:

   ```
   kumpcon validate teps2.mdl
   ```

   Example 5-13 shows the **kumpcon validate** output.

Example 5-13 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 used Take Action commands to import the metafile.

The **validate** command generates a report that has the same name as the metafile, but with the .rpt extension, showing the application definition. Example 5-14 shows the output report for teps2.mdl metafile.

Example 5-14 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
```

In this scenario, we used a Take Action command in the Tivoli Enterprise Portal to import the metafile:

1. Right-click the ODBC data provider managed system, and select **Take Action** → **Select**, as shown in Figure 5-87.

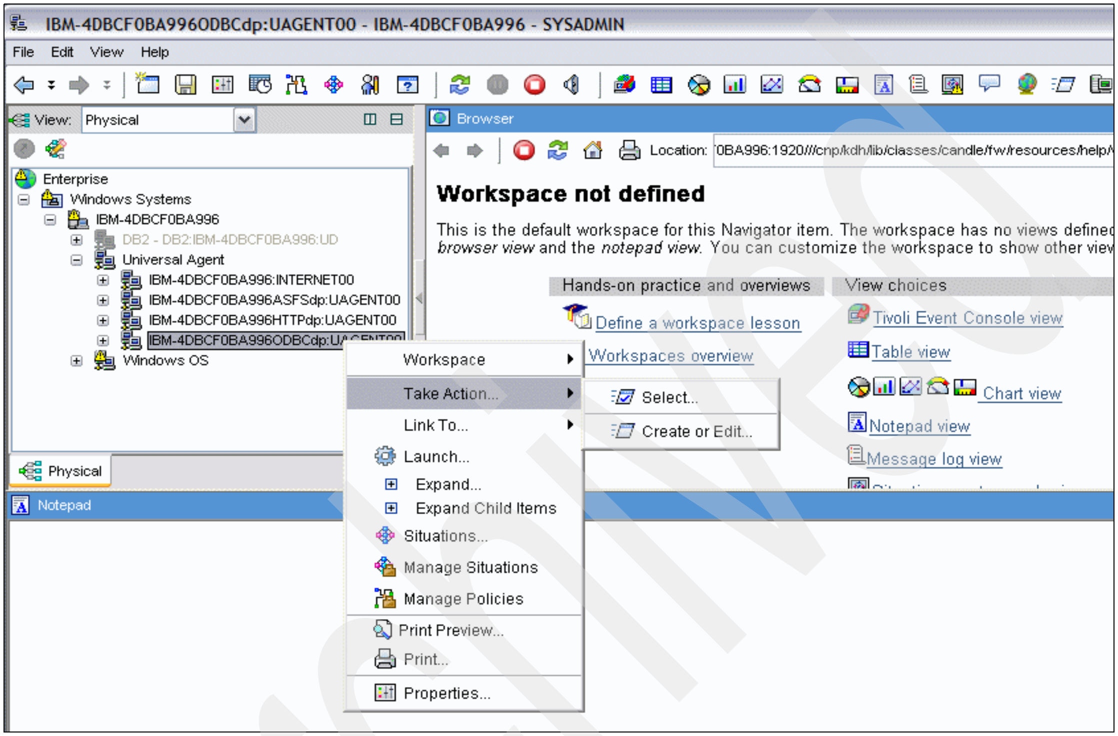


Figure 5-87 Importing the ODBC metafile using Take Action

2. In the Take Action window, select **Control Import** and enter argument values (Figure 5-88).

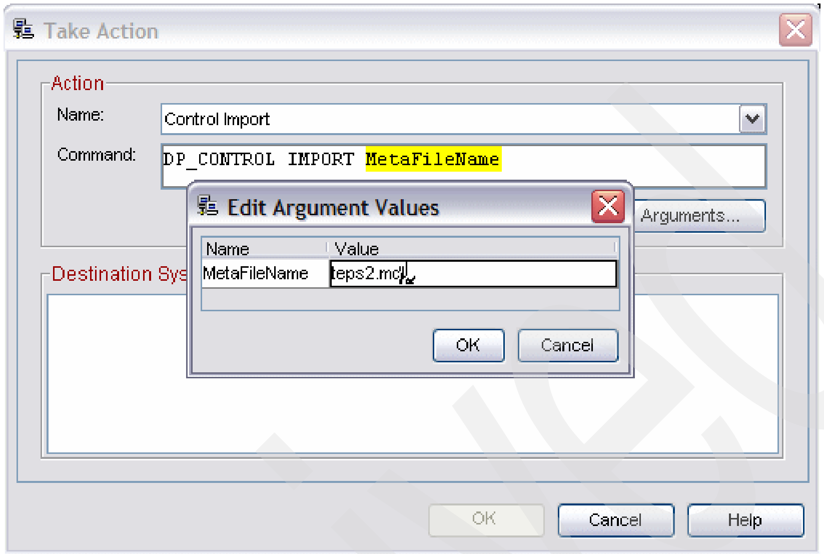


Figure 5-88 Select Control Import and enter argument values

3. Select the managed system to execute the action and click **OK** (Figure 5-89).

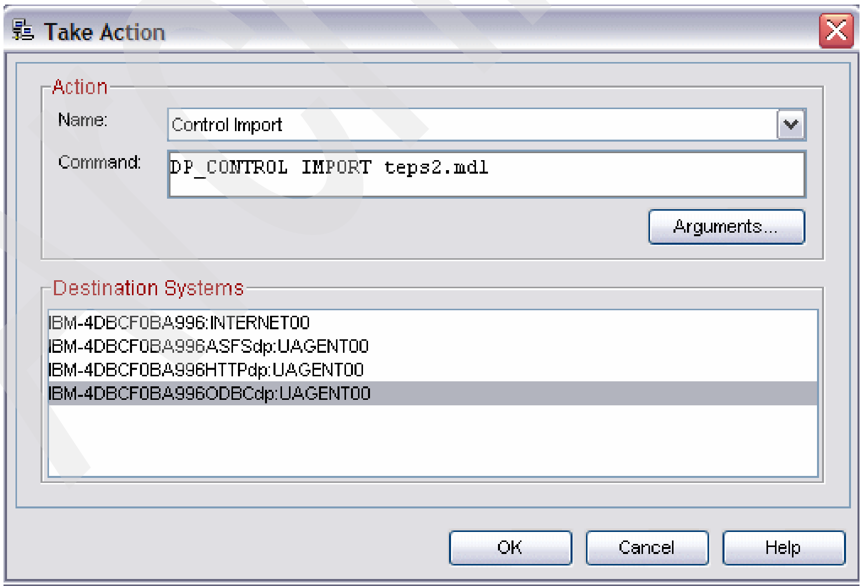


Figure 5-89 Selecting the managed system to execute the action

4. Figure 5-90 shows the new application of the metafile with two attribute groups (KFWLOGIN and KFWUSER) after the import.

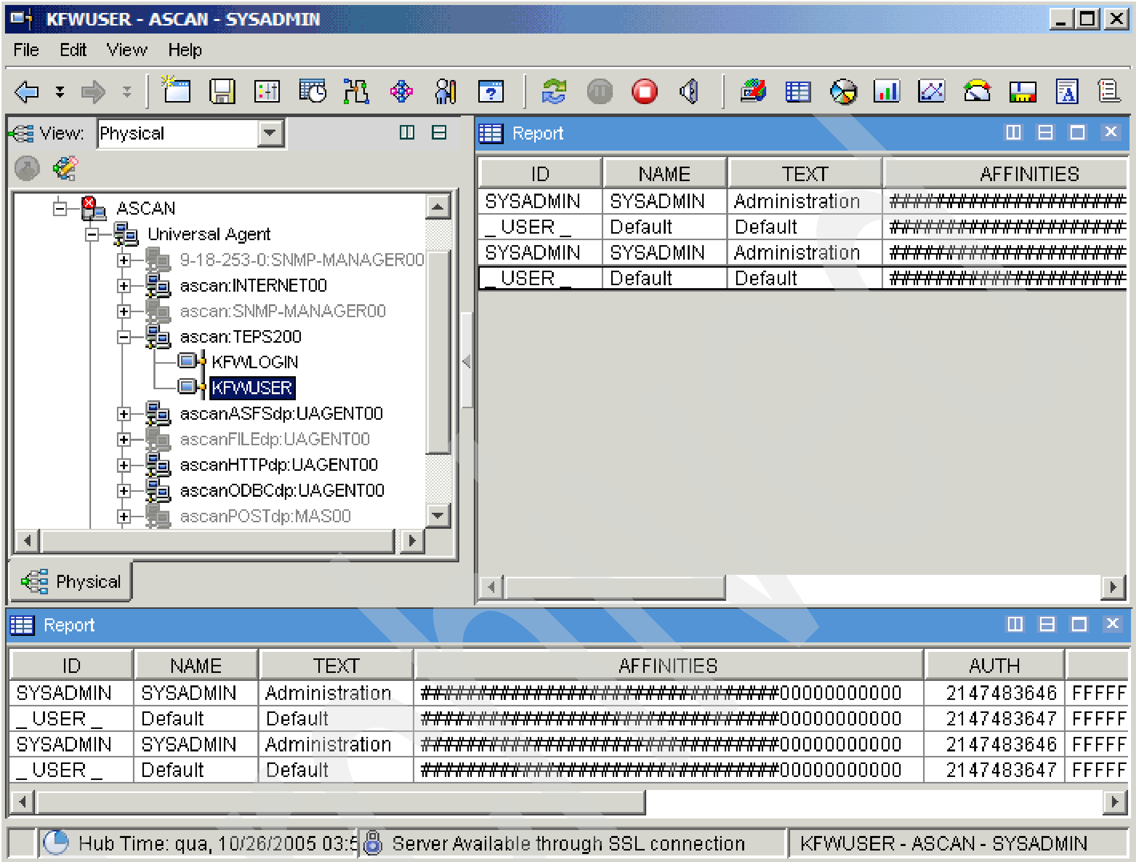


Figure 5-90 New attribute groups with the same data

5. 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 5-15 shows the metafile after the changes in the //NAME parameter.

Example 5-15 //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
```

6. After changing the statement and saving the metafile, refresh it in the Universal Agent. Enter the following command to refresh the metafile:

   `kumpcon refresh teps2.mdl`

   Example 5-16 shows the output of the **refresh** command on the teps2.mdl metafile.

Example 5-16 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.
```

The data in Tivoli Enterprise Portal will now be correct, as shown in Figure 5-91.

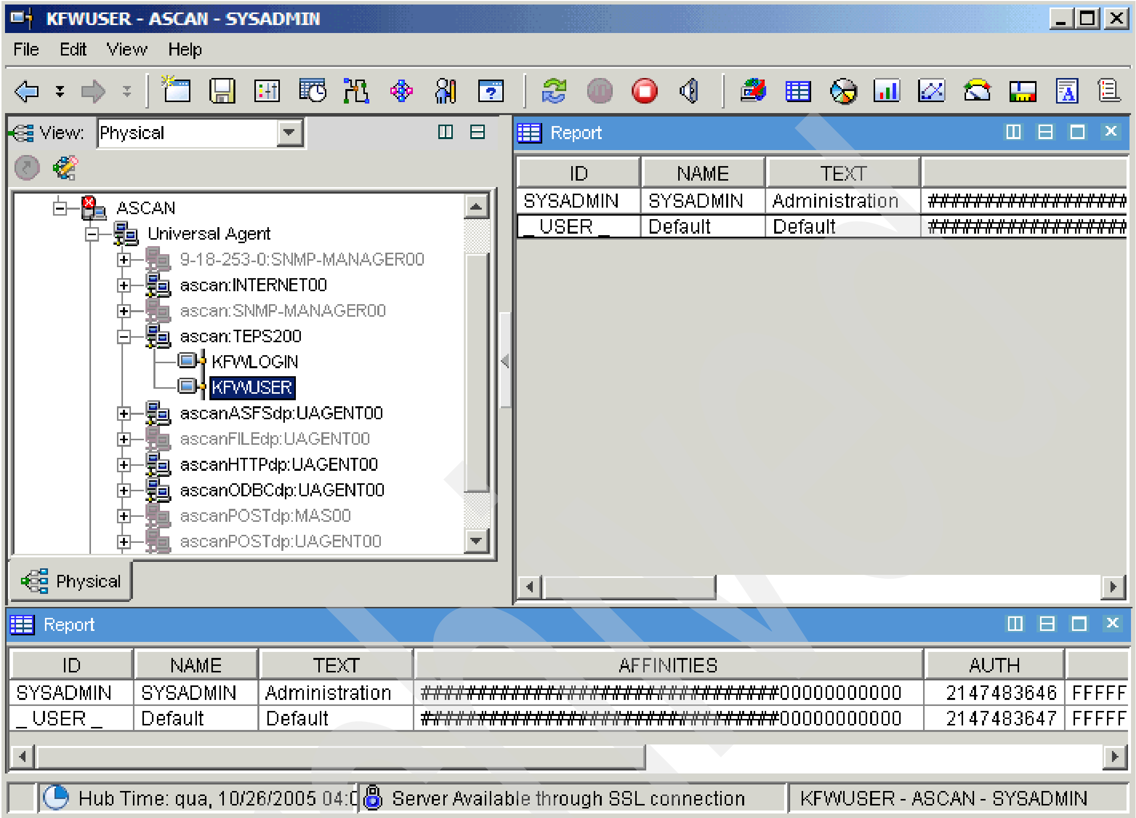


Figure 5-91 Data in Tivoli Enterprise Portal after changes in the metafile

5.13 SOAP scenarios

SOAP is a way for a programs running on the same or different operating systems to communicate with each other. In this section, we describe some scenarios to monitor SOAP using IBM Tivoli Monitoring Express V6.1.

5.13.1 What is Tivoli Enterprise Monitoring Web Services?

The Tivoli Enterprise Monitoring Web Services solution provides you with an industry-standard open interface into IBM Tivoli Monitoring Express V6.1 solutions. This open interface provides easy access to Tivoli performance and availability data, enabling you to use this information for advanced automation and integration capabilities. Tivoli Enterprise Monitoring Web Services implements a client/server architecture. The client sends SOAP requests to the SOAP server. The server receives and processes the SOAP requests from the client.

Predefined SOAP methods let you perform many functions within the monitored environment. You can begin to use the SOAP methods immediately. You can also use these SOAP methods as templates in creating your own advanced methods. SOAP works with any programming or scripting language, any object model, and any Internet wire protocol. Tivoli SOAP methods can be invoked through Perl, JavaScript™, VBScript, JScript®, C++, and a browser.

What is SOAP?

SOAP is a communications XML-based protocol that lets applications 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.

> **Important:** Prior to using the Tivoli solution, you must have a basic understanding of SOAP, of Extensible Markup Language (XML) and XML namespaces, and of the Web Services Description Language (WSDL). To access information and tutorials about these topics, refer to the following sites:
>
> `http://w3schools.com`
> `http://w3.org/TR/SOAP`

5.13.2 Using Tivoli Enterprise Monitoring Web Services

Tivoli provides numerous SOAP methods with Tivoli Enterprise Monitoring Web Services. These methods enable you to dynamically query and control Tivoli Monitoring environments. Using the Tivoli SOAP methods, you can:

- Stop or start situations
- Forward IBM Tivoli 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 service to test a request to ensure that it works properly. You can then create a policy (requires Enterprise Edition of IBM Tivoli Monitoring) that submits multiple requests for processing. In addition, you can generate daily operation summaries. You can retrieve data in the Tivoli Data Warehouse, as described in 4.3, “Historical data collection” on page 191.

Note: Tivoli Enterprise Monitoring Web Services provides XML data rows. Use the Tivoli SOAP methods in combination with your own scripts to display the data in chart and tables.

User IDs

Important: At installation and configuration time, you will be asked to supply user IDs for those who need access to monitoring server data. If no user IDs are supplied, all users will be given permission to update data.

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. You can also make changes at a later time to add or to remove users’ access to monitoring server data. See “Adding users” on page 340 for details.

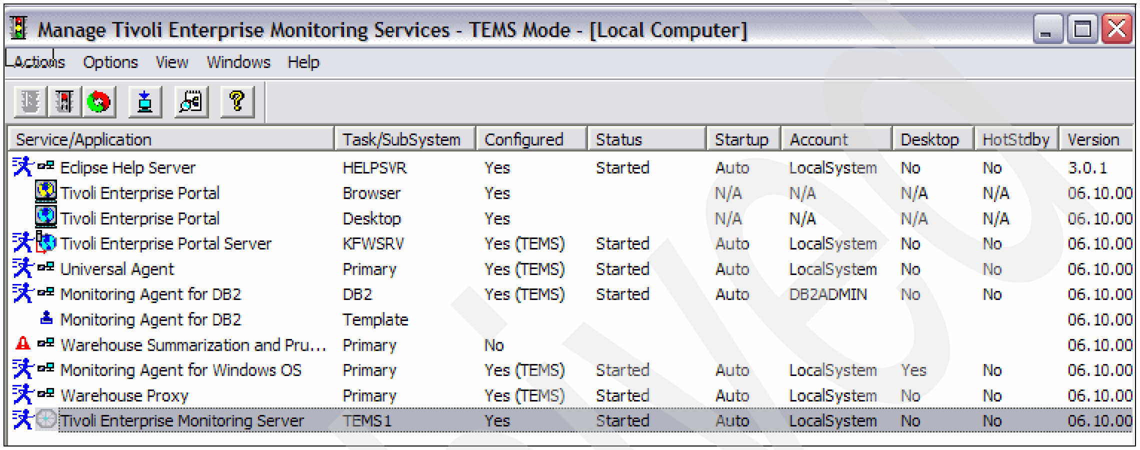
5.13.3 Starting the SOAP client and making a request

The following steps describe the process of starting the SOAP client and making a request.

Configuring Tivoli Enterprise Monitoring Server

In this step, you use the Manage Tivoli Enterprise Monitoring Services window to activate the SOAP server and define hubs with which the SOAP server can communicate. Follow these steps:

1. Open the Manage Tivoli Enterprise Monitoring Services window (Figure 5-92).



Service/Application	Task/SubSystem	Configured	Status	Startup	Account	Desktop	HotStdby	Version
Eclipse Help Server	HELPSVR	Yes	Started	Auto	LocalSystem	No	No	3.0.1
Tivoli Enterprise Portal	Browser	Yes		N/A	N/A	N/A	N/A	06.10.00
Tivoli Enterprise Portal	Desktop	Yes		N/A	N/A	N/A	N/A	06.10.00
Tivoli Enterprise Portal Server	KFWSRV	Yes (TEMS)	Started	Auto	LocalSystem	No	No	06.10.00
Universal Agent	Primary	Yes (TEMS)	Started	Auto	LocalSystem	No	No	06.10.00
Monitoring Agent for DB2	DB2	Yes (TEMS)	Started	Auto	DB2ADMIN	No	No	06.10.00
Monitoring Agent for DB2	Template							06.10.00
Warehouse Summarization and Pru…	Primary	No						06.10.00
Monitoring Agent for Windows OS	Primary	Yes (TEMS)	Started	Auto	LocalSystem	Yes	No	06.10.00
Warehouse Proxy	Primary	Yes (TEMS)	Started	Auto	LocalSystem	No	No	06.10.00
Tivoli Enterprise Monitoring Server	TEMS1	Yes	Started	Auto	LocalSystem	No	No	06.10.00

Figure 5-92 Configure SOAP interface

2. Right-click **Tivoli Enterprise Monitoring Server** and select **Advanced** → **Configure SOAP Hubs**.

   The SOAP Server Hubs Configuration window opens (Figure 5-93).

3. Click **Add Hub**. The Hub Specification window opens.

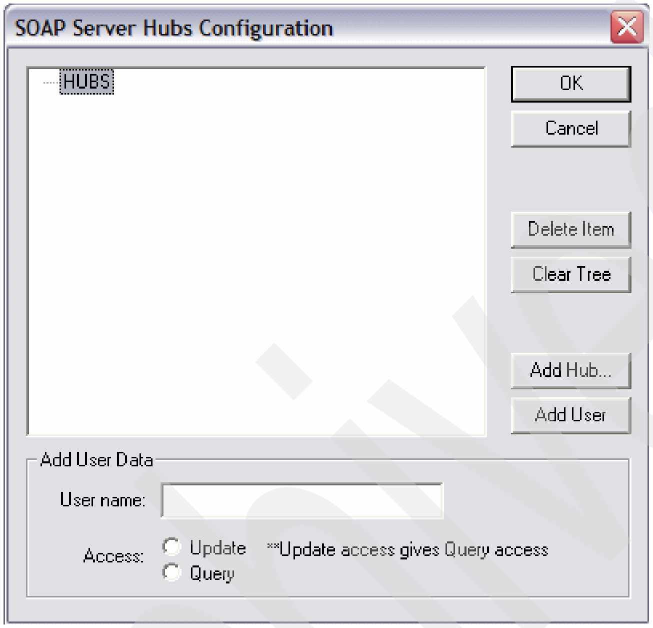


Figure 5-93 SOAP Server Hubs Configuration

4. Select the communications protocol from the Protocol drop-down list.
5. Specify an alias name in the Alias field (for example “SOAP”).
6. Take one of the following actions:
   – If you are using TCP/IP or TCP/IP PIPE communications, specify the values shown in Table 5-18.

Table 5-18 TCP/IP fields in Hub Specification window

Field	Description
Hostname or IP Address	The host name or TCP/IP address of the host machine
Port	The TCP/IP listening port for this host

– If you are using SNA communications, specify the values shown in Table 5-19.

Table 5-19 SNA fields in Hub Specification window

Field	Description
Network name	Your site’s SNA network identifier.
LU name	The LU name for the Tivoli Enterprise Monitoring Server. This LU name corresponds to the Local LU Alias in your SNA communications software.
LU6.2 LOGMODE	The name of the LU6.2 logmode. Default is CANCTDCS.
TP Name	The Transaction Program name for the Tivoli Enterprise Monitoring Server.

7. Click **OK**.

   The server tree opens (Figure 5-94).

Figure 5-94 SOAP configuration: Server tree

Adding users

In this step, you define users on the hub and specify user access rights (query or update). Follow these steps:

1. Select the server (click anywhere within the server tree), if necessary.
2. In Add User Data, enter the user name.

   User IDs must be identical to those specified for Tivoli Enterprise Monitoring Server logon validation. Access is restricted to only that hub monitoring server to which a user has access.

   Note: If no user IDs are supplied, all users will be given permission to update data.

3. Select the type of user access (Query or Update).
4. Click **Add User**.

   The server tree is updated, showing the user and type of access (Figure 5-95).

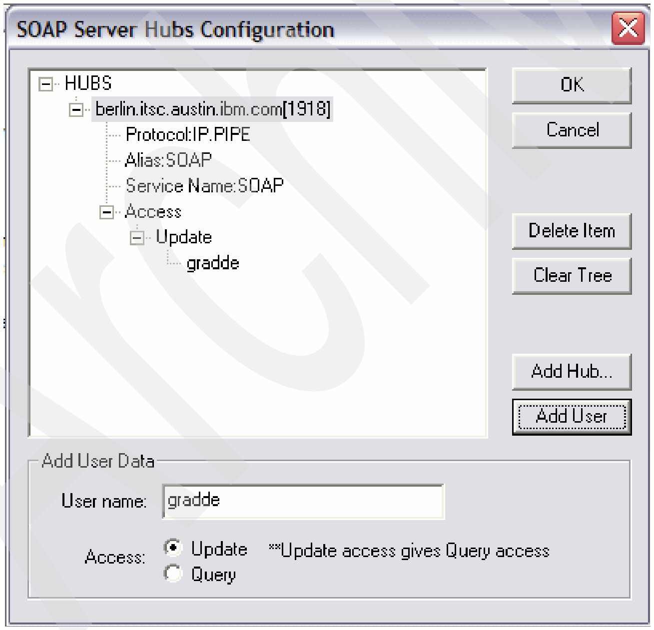


Figure 5-95 New user added to the server tree

5. To delete a user, select (highlight) the user name from the tree and click **Delete Item**.

6. To delete a hub, click anywhere within the hub's tree and click **Clear Tree**.

Verifying the configuration

There are several ways of starting the Tivoli SOAP client. We describe two ways here:

- Using Internet Explorer
- Using SOAP client command line utility

When you use the SOAP client in conjunction with Microsoft Internet Explorer to issue SOAP requests, you can modify, if needed, the tags or the text. In contrast, the command line utility simply displays the output of the requests at the command prompt.

> **Note:** Before you can access newly created Universal Agent objects, the hub monitoring server where the SOAP server is running must be recycled.

Using Internet Explorer

Perform the following steps:

1. Start Internet Explorer. Be sure to enable the Access data sources across domains in Internet Explorer's security settings.
2. In the Address field, type the URL for SOAP client:

   ```
   http://localhost:1920///cms/soap/kshhsoap.htm
   ```

   Where localhost can be used literally when accessing the SOAP server running on the same computer or changed to the proper host name or network address of a SOAP server running on a different computer. The port number for the Tivoli HTTP service is 1920.

   > **Note:** You can also route requests to a remote hub by replacing `soap` in the Address field with the alias name of the hub you want to access (`HUB_localhost` in the following example). The alias must have been previously defined to the SOAP server. For example:
   >
   > `http://localhost:1920///cms/HUB_localhost/kshhsoap.htm`

The SOAP client HTML page opens (Figure 5-96).

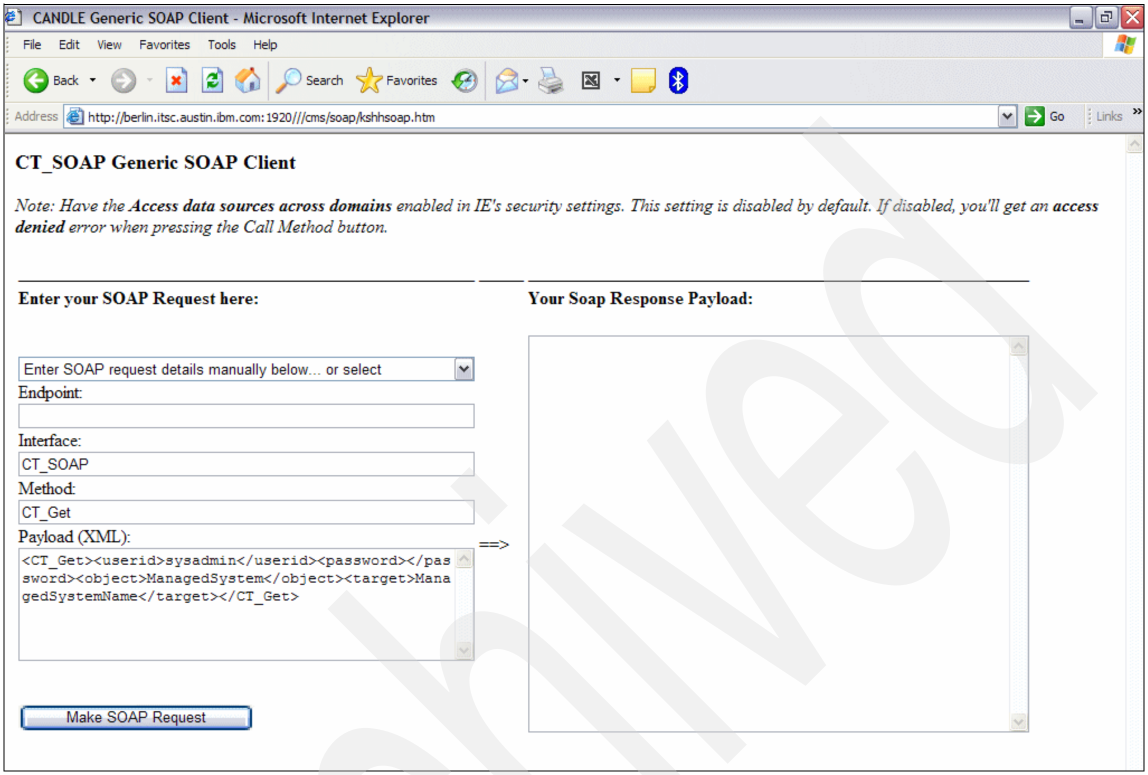


Figure 5-96 SOAP client HTML page

3. Select a SOAP method from the list in the first field. After you select a method, the other fields fill in automatically.
4. Modify, if needed, the tags or the text in the “Edit Payload (XML)” area.
5. Click **Make SOAP Request**. The output of the request displays in the Your SOAP Request Payload area.

> **Note:** When issuing a CT_Get request against a particular agent type, the monitoring server where the SOAP server is running must be configured as needed for that agent type. For example, when issuing CT_Get request for a UNIX agent connected to a UNIX monitoring server, the monitoring server running the SOAP server must be configured and seeded for that UNIX agent.

Using the SOAP client command line utility kshsoap

Perform the following steps:

1. Open a DOS window.
2. Change to the c:\IBM\ITM\CMS directory.
3. In the current directory, create a Notepad file named “SOAPREQ.txt” containing the following SOAP request:

   ```
   <CT_Get><object>ManagedSystem</object></CT_Get>
   ```

   Or if security has been enabled:

   ```
   <CT_Get><userid>logonid</userid><password>password</password><object>Manage
   dSystem</object></CT_Get>
   ```

4. Create another Notepad file named “URLS.txt” containing URLs that receive the SOAP request. For example:

   ```
   http://hostname:1920///cms/soap
   ```

5. Enter this command:

   ```
   kshsoap SOAPREQ.txt URLS.txt
   ```

   (SOAPREQ.txt is the name of the file that contains the SOAP request and URLS.txt is the name of the file that contains the URLs).

The **kshsoap** utility processes the SOAPREQ.txt file and displays the output of the SOAP request in the DOS window. The SOAP request is sent to each URL listed in the URLS.txt, and the SOAP response from each URL displays in the DOS window.

Issuing SOAP requests as system commands

In Tivoli Enterprise Portal, you can use the Take Action feature to issue SOAP requests as system commands in policies or in situations. The SOAP requests are stored in a text file. In Tivoli Enterprise Portal, you can issue a SOAP request in a situation using the Action tab of the Situation Editor.

The SOAP command is:

```
soap:CT_Execute,filename=SOAPREQ
```

Where:

- CT_Execute is the name of the SOAP method that enables you to execute a SOAP request that is stored in a file.
- SOAPREQ is the name of the file you created that contains the CT_EMail SOAP request.

For example, the SOAPREQ file might contain the content shown in Example 5-17.

Example 5-17 SOAPREQ

```
<CT_EMail><server>n-smtpmta</server><sender>soap@ibm.com</sender><receiver>haak
an.gradin@uk.ibm.com</receiver>untouched by human
hands</subject><attachmenttitle>AFData.htm</attachmenttitle><request><attach>re
s.pfx</attacid=__XMLID__ ><CT_Redirect
endpoint=__http://sp22.ibm.com:18882__><SOAP-ENV:Envelope
xmlns:SOAP-ENV=__http://schemas.xmlsoap.org/soap/envelope/__
xmlns:soap=__http://schemas.xmlsoap.org/wsdl/soap/__ > <SOAP-ENV:Body>
<AF_Execute> <Exec>SOAP0002</Exec> </AF_Execute> </SOAP-ENV:Body>
</SOAP-ENV:Envelope></CT_Redirect></request><request><attach>res.sfx</attach></
request></CT_EM
```

5.13.4 Tivoli Enterprise Monitoring Web Services scenarios

Here are a few examples of how you might use Tivoli Enterprise Monitoring Web Services. You can use these examples as suggestions for creating your own applications.

> **Note:** These scenarios do not describe the actual code that was used to develop them. To produce the charts and tables shown in these examples, you need to develop your own scripts.

Generating daily business operation summaries and charts

You can retrieve data from multiple agents, using the SOAP server against a live hub, to generate daily business operation summaries. You can use the CT_EMail SOAP method to e-mail these summaries to management.

You might want to add an <insert> tag into CT_EMail. This tag contains instructions for the preferred format for the summaries.

Management can view these summaries at their desktops using Internet Explorer, thereby removing the need to install and launch the OMEGAMON platform user interface. Summaries provide an efficient and speedy look at problems that might occur during the night.

In addition to the general features you can add to tables and charts, charts and tables for these summaries might contain these features:

- Transaction volumes/response times and whether they are meeting service levels can be plotted with respect to resource trends and error conditions.

- Charts can be plotted over multiple segments, making them easier to view and to print.
- The x-axis can use a variable scale to throw the prime shift in greater detail.
- Multiple objects/attributes can be plotted from multiple sources and exceptions can be correlated by time, providing focus on problem areas.
- A status map can show the status of situations.

Obtaining data snapshots and offline table and charts

Using the SOAP method CT_Get against a live hub, you can obtain a data snapshot from multiple agents to produce charts and reports. You can also create an AF/REMOTE REXX script that requests a snapshot of its data.

In addition to the general features you can add to tables and charts, charts and tables for this type of request might contain these features:

- The chart can be plotted over multiple segments, making it easier to view and print.
- Clicking the attribute name in the legend box displays that attribute in the y-axis and shows its threshold value.
- The threshold value, when changed, can be used as the new threshold value.

Figure 5-97 and Figure 5-98 on page 346 show sample charts/reports generated for this type of request.

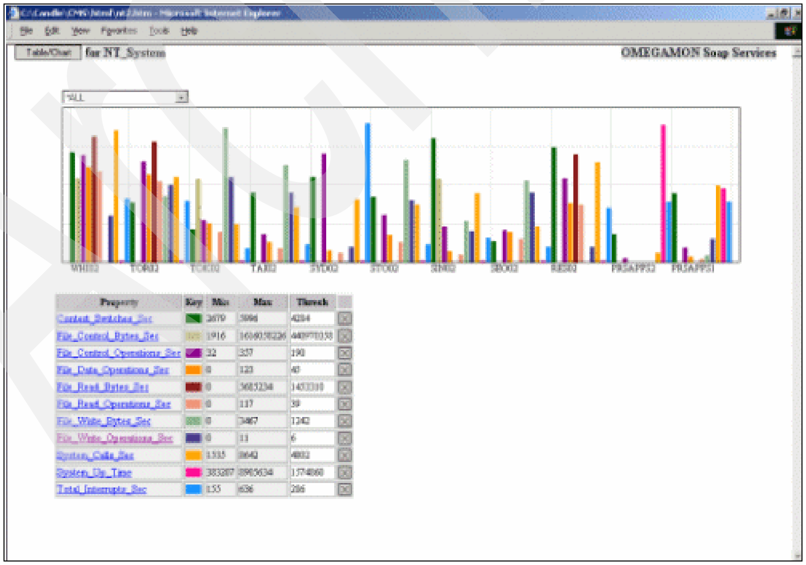
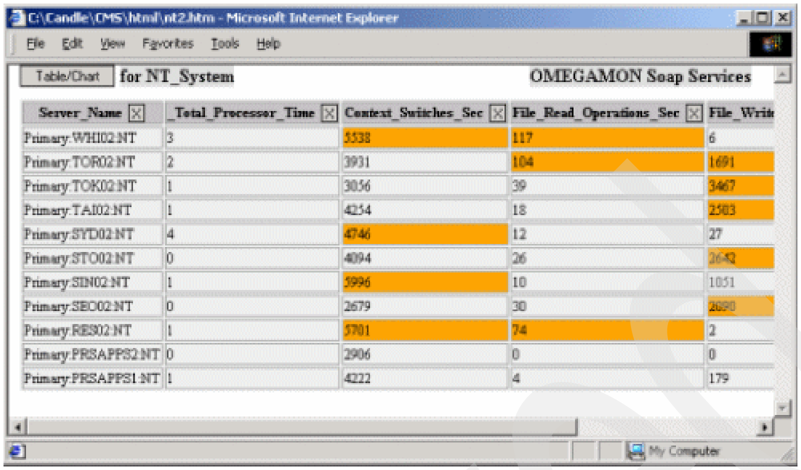


Figure 5-97 Data snapshot chart and table



Server_Name	_Total_Processor_Time	Context_Switches_Sec	File_Read_Operations_Sec	File_Writ
Primary:WHI02:NT	3	5538	117	6
Primary:TOR02:NT	2	3931	104	1691
Primary:TOK02:NT	1	3056	39	3467
Primary:TAI02:NT	1	4254	18	2503
Primary:SYD02:NT	4	4746	12	27
Primary:STO02:NT	0	4094	26	2642
Primary:SIN02:NT	1	5996	10	1051
Primary:SEO02:NT	0	2679	30	2690
Primary:RES02:NT	1	5701	74	2
Primary:PRSAPPS2:NT	0	2906	0	0
Primary:PRSAPPS1:NT	1	4222	4	179

Figure 5-98 Data snapshot table

Sending alerts into an OMEGAMON platform

Using the SOAP method CT_Alert, you can send a new alert into an OMEGAMON platform. For example, AF/REMOTE detects a problem on a Tandem system and generates an alert in an OMEGAMON platform. The OMEGAMON platform then displays alert information from the Tandem platform.

Collaborative automation using AF/REMOTE

You can create an AF/REMOTE REXX application that calls JScript SOAP functions to forward any AF/REMOTE trapped message and display it on a Universal Message console. You can use AF/REMOTE scripts to trap and send any log messages, console messages, and so on, to Tivoli Enterprise Monitoring Server using SOAP methods.

You can create an application that provides these benefits:

- You can monitor devices, such as Tandem, by trapping VT100 messages and raising Universal Messages.
- You can send commands to AF/REMOTE monitored Telnet sessions and send replies back to these commands.
- Source messages can be either excluded or included, based on any criteria using powerful regular expressions.
- A local log can keep audit information about the status of messages received.
- A local log can keep information about the source hub connection/retry status.

Figure 5-99 and Figure 5-100 show a sample Telnet session, a Universal Message console showing messages received, and a sample message log.

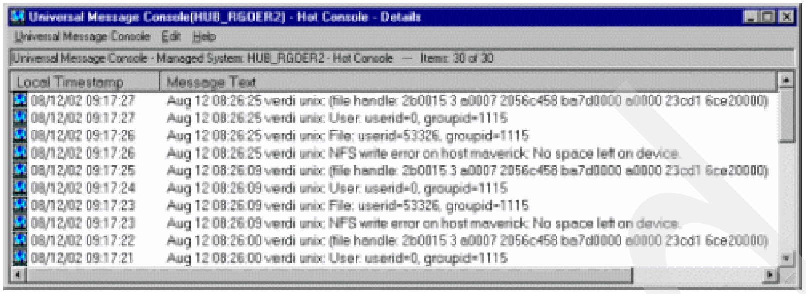


Figure 5-99 Universal Message Console showing messages received

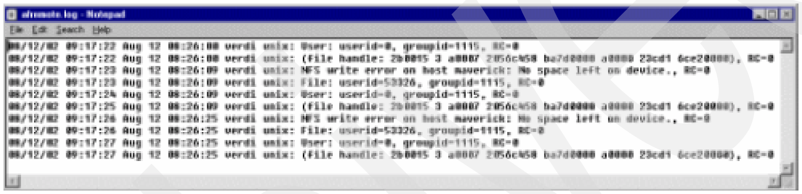


Figure 5-100 Message log details

Acknowledging an event on an OMEGAMON platform

You can acknowledge an event on an OMEGAMON platform. In this example, we use an OMEGAMON automation tool called AF/OPERATOR®:

1. AF/OPERATOR receives a raised situation from an OMEGAMON platform.
2. AF/OPERATOR pages a responsible party who, in turn, sends back an acknowledgement.
3. AF/OPERATOR forwards the acknowledgement of the alert to the OMEGAMON platform.

To accomplish this task, use the CT_Acknowledge SOAP method. This method gives you the ability to control events on an OMEGAMON platform based on information obtained and detected by the Tivoli automation solutions.

Report contents

You can design a report to contain both a table and a chart view. You might want to add a Table/Chart button that enables you to toggle between the chart and the table view.

Chart view features

Charts can have specific features. For example, you can design charts that enable you to:

- View different types of charts, depending on the data retrieved.
- Choose the y-axis by selecting additional attributes from the drop-down attribute list.
- Change the title and instructions for the chart.
- View the flyover text containing the name and value of the attribute plotted by placing your cursor over each plotted item.

Table view features

Tables can have specific features. For example, you can design tables that enable you to:

- View the flyover text containing the name and value of the attribute plotted by placing your cursor over each plotted item.
- Remove attributes from a table by clicking the X button next to the attribute name.

6

Troubleshooting

This chapter introduces you to the troubleshooting components of the product set. This product set is 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 you can perform to troubleshoot and correct problems with the product. It is not exhaustive, and in some cases, IBM Support is required to isolate issues and correct them.

In this chapter, we discuss the following topics:

- Overview
- Message logging
- Trace facilities
- Using the product documentation
- Sample problem scenarios
- Common installation problems in IBM Tivoli Monitoring Express V6.1
- Tivoli Enterprise Portal Server hints
- Tivoli Enterprise Monitoring Server hints
- Tivoli Enterprise Portal troubleshooting
- IBM Tivoli Universal Agent troubleshooting
- Working with IBM Support

6.1 Overview

This chapter provides guidelines for efficient troubleshooting of the IBM Tivoli Monitoring Express V6.1 product. Troubleshooting can be defined as the process of identifying, isolating, and ultimately resolving problems. In the case of IBM Tivoli Monitoring Express V6.1, you can perform several steps to go through the troubleshooting process.

Identifying the problem is the first step and 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.

6.2 Message logging

Each component of IBM Tivoli Monitoring Express V6.1, from Tivoli Enterprise Monitoring Server and Tivoli Enterprise Portal Server to the agents, has messaging facilities to provide feedback about exceptional issues that occur. The messages can be informational, warning, or error messages in nature depending on the message. The message output destination can be the screen or a log file. These messages are documented in *IBM Tivoli Monitoring Problem Determination Guide V6.1*, GC32-9458.

Some samples of the three types of messages include:

- Informational: `KFWITM006I Validating user credentials`
- Warning: `KFWITM197W User has no assigned Navigator Views`
- Error: `KFWITM215E Unable to process logon request`

6.3 Trace facilities

IBM Tivoli Monitoring Express V6.1 contains an extensive trace facility that can provide helpful information about the state of the components. IBM Tivoli Monitoring Express V6.1 creates several types of logs, and the principal log type is the reliability, availability, and serviceability (RAS) log. RAS logs are in English and are available on the Tivoli Enterprise Monitoring Server, the Tivoli Enterprise Portal Server, and the monitoring agents. Other logs include installation, seed, LG0, Open Database Connectivity (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 6-1 summarizes the IBM Tivoli Monitoring Express V6.1 log names and locations.

Table 6-1 Log names and locations

Windows		**UNIX**	
TEPS	*ITM_InstallDir*\logs*hostname_cq_timestamp-xx*.log	TEPS	Currently not available on UNIX.
TEMS	*ITM_InstallDir*\logs*hostname*_ms_*timestamp-xx*.log	TEMS	Currently not available on UNIX.
Agents	*ITM_InstallDir*\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 monitoring server, situations running, and the status of Take Actions.	Agents	ITM_InstallDir/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 monitoring server, situations running, and the status of Take Actions.
Warehouse Proxy	*ITM_InstallDir*\logs*hostname*_hd_*timestamp-xx*.log	Warehouse Proxy	Currently not available on UNIX.
tacmd	*ITM_InstallDir*\bin\kuiras1.log	tacmd	*ITM_InstallDir/logs/kuiras1.log*
Seeding process	*ITM_InstallDir*\CNPS\logs\seedPPC.log *ITM_InstallDir* installITM\logs\CMSSeed.log for the monitoring server	Seeding process	*ITM_InstallDir*/logs/*hostname*_ci_<*tems pid*>.log
Summarization and Pruning agent	*ITM_InstallDir*/logs/*hostname_sy_timestamp-xx*.log	Summarization and Pruning agent	Currently not available on UNIX.

Windows		UNIX	
Installation logs	*ITM_InstallDirITM_InstallDir*\Install ITM ► Main installation log: IBM Tivoli Monitoring Express *date PID*.log ► Configuration of the Tivoli Enterprise Portal Server ODBC connection: TEPS_ODBC.log ► Configuration of the Warehouse Proxy: Warehouse_Configuration.log	Installation logs	Currently not available on UNIX.

> **Note:** In Table 6-1, we define the following variables:
>
> - xx: The rotating log number
> - PC: The two-letter product code (for example, LZ for the Linux agent)
> - PPC: The three-letter product code (for example, KLZ for the Linux agent)

6.3.1 Trace settings

The tracing of components is controlled by several environment variables, and there are several methods by which you can modify these variables. Table 6-2 defines some of the environment variables. (This is not an exhaustive list.)

Table 6-2 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	Log file location of the RAS1 log.
INVENTORY	File containing the inventory of RAS1 logs for the component.
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.

Variable	Description
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_ENTERPRISE=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 previously default to No. As an example, if you use the SNMP data provider and have problems collecting MIB data, you set the following 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) (UNIT:kumpdpda Error Output)	Detailed File data provider tracing.
ERROR (UNIT:kump ALL)	Problems involving all data provider processing.
ERROR (UNIT:kumaeagt ALL) (UNIT:kumpemit ALL)	SNMP Emitter processing.
ERROR (UNIT:kumamain ALL) (UNIT:kumfaagt ALL) (UNIT:kumadtl ALL) (UNIT:kumpdpda Error Output)	Non-firing situations tracing.
ERROR (UNIT:kumpsosr ALL) (UNIT:kumpspst ALL) (UNIT:kumpscku ALL) (UNIT:kumpstcp ALL) (UNIT:kumplpba ALL) (UNIT:kumpdpda Error Output)	Detailed application programming interface (API) or socket data provider tracing.
ERROR (UNIT:kumpcadm ALL)	Metafile parsing tracing.
ERROR (UNIT:kumamain ALL)	Problems involving managed system online/offline processing.
ERROR (UNIT:kumamain ALL) (UNIT:kumfaagt ALL) (UNIT:kumadtl ALL) (UNIT:kumpdpda Error Output)	Problems involving Universal Agent request processing.
ERROR METRICS	Problems involving Universal Agent memory usage.

The following exercise shows you how to use the Tivoli Enterprise Monitoring Services console to manually modify the appropriate environment files:

1. On the Windows system where you installed the product, click **Start** → **Programs** → **IBM Tivoli Monitoring** → **Manage Tivoli Monitoring Services**. The Manage Tivoli Enterprise Monitoring Services window opens, as shown in Figure 6-1.
2. Modify the appropriate environment file:
   – Tivoli Enterprise Monitoring Server: Highlight **IBM Tivoli Enterprise Monitoring Server** and click the **Stop** button (red traffic light) on the toolbar. Modify the environment KBBENV file by selecting **Advanced** → **Edit ENV File** (Figure 6-1) and click the **Start** button (green traffic light) on the toolbar. The file's location is ITM_InstallDir\CMS\KBBENV.

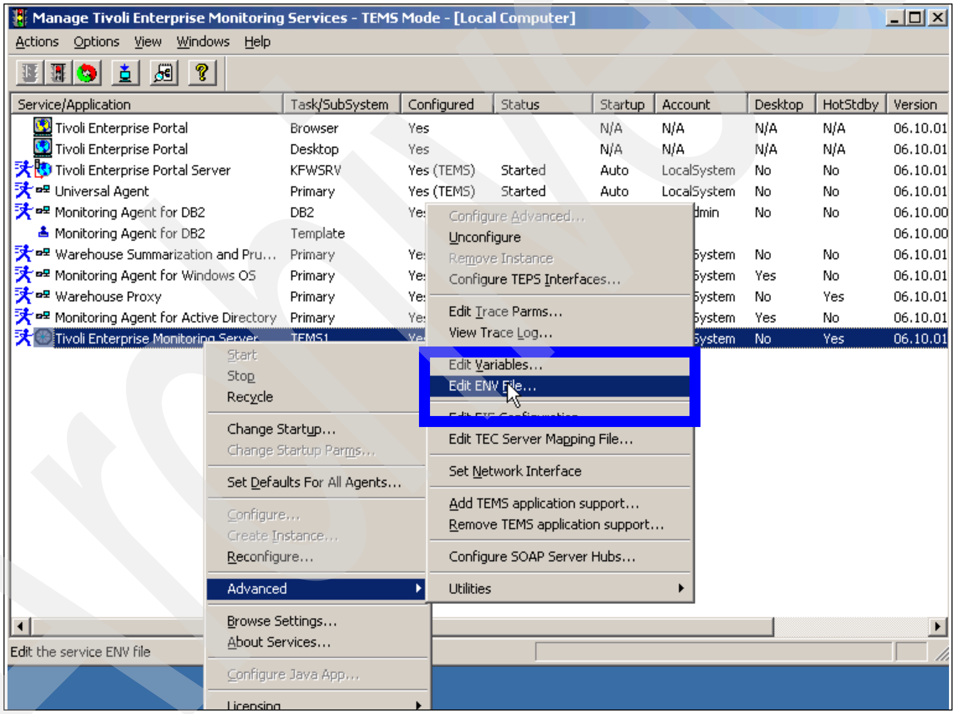


Figure 6-1 Modifying the KBBENV file

– Tivoli Enterprise Portal Server: Highlight **IBM Tivoli Enterprise Portal Server** and click the **Stop** button (red traffic light) on the toolbar. Modify the environment KFWENV file by selecting **Advanced** → **Edit ENV File** (Figure 6-2) and click the **Start** button (green traffic light) on the toolbar. The file's location is ITM_InstallDir\CNPS\KFWENV.

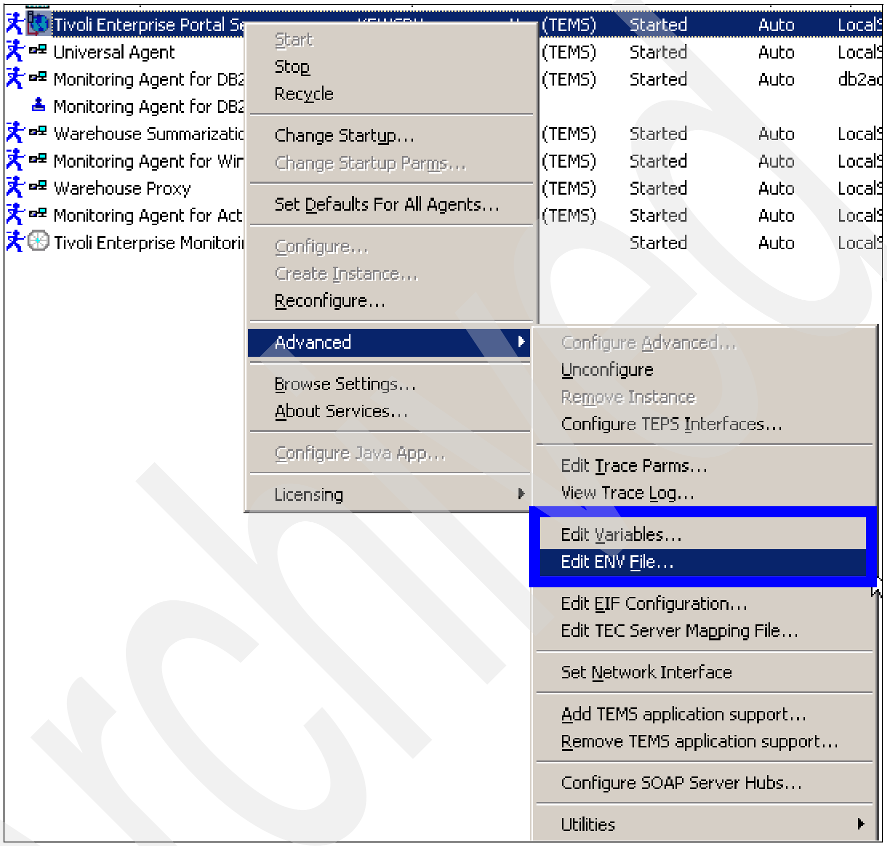


Figure 6-2 Modifying the KFWENV file

– Tivoli Enterprise Monitoring Agents on Windows are located in *ITM_InstallDir*\TMAITM6*PPC*ENV (where *PPC* is the three-letter product code for the agent). For example, highlight the **Monitoring Agent for Windows OS** and select the **Stop** button (red traffic light) on the toolbar. Modify the environment KNTENV file by selecting **Advanced** → **Edit ENV File** and click the **Start** button (green traffic light) on the toolbar. The file's location is <ITM_InstallDir>\TMAITM6\KNTENV.

Tivoli Enterprise Monitoring Agents on UNIX are located in *ITM_InstallDir/*config/*pc*.config (where *pc* is the two-letter product code for the agent). For example, modify the configuration file /opt/IBM/ITM/config/lz.config.

– Command lines such as **tacmd** can be logged in:

  - Windows: ITM_InstallDir\bin\KUIENV
  - UNIX: ITM_InstallDir\bin\tacmd

6.3.2 Identifying the latest log files

Because of the log-rolling mechanism, it can be difficult to determine which is the most recent log file. You can easily determine this by consulting the log file information file, with the format as shown in Figure 6-3.

```
03/18/2006  11:29 PM        5,191,007 BERLIN_ms_441a201f-01.log
03/18/2006  08:06 PM        5,191,202 BERLIN_ms_441a201f-02.log
03/18/2006  11:29 PM        1,414,655 BERLIN_ms_441a201f-03.log
03/22/2006  12:35 PM        5,191,537 BERLIN_ms_441effcf-01.log
03/22/2006  12:21 PM        5,191,688 BERLIN_ms_441effcf-02.log
03/22/2006  12:35 PM          102,646 BERLIN_ms_441effcf-03.log
03/23/2006  12:48 AM        5,191,596 BERLIN_ms_4421b521-01.log
03/24/2006  01:40 PM        5,191,753 BERLIN_ms_4421b521-02.log
03/24/2006  01:40 PM          815,645 BERLIN_ms_4421b521-03.log
```

Figure 6-3 List log for Tivoli Enterprise Monitoring Server on Windows

In the Tivoli Enterprise Portal Server log information file (Figure 6-4), the top entry is the one currently being written to:

```
03/24/2006  01:40 PM              468 BERLIN_ms.inv
```

```
<ITM_InstallDir>\logs\BERLIN_ms_4421b521-03.log
<ITM_InstallDir>\logs\BERLIN_ms_4421b521-02.log
<ITM_InstallDir>\logs\BERLIN_ms_4421b521-01.log
<ITM_InstallDir>\logs\BERLIN_ms_441effcf-03.log
<ITM_InstallDir>\logs\BERLIN_ms_441effcf-02.log
<ITM_InstallDir>\logs\BERLIN_ms_441effcf-01.log
<ITM_InstallDir>\logs\BERLIN_ms_441a201f-03.log
<ITM_InstallDir>\logs\BERLIN_ms_441a201f-02.log
<ITM_InstallDir>\logs\BERLIN_ms_441a201f-01.log
```

Figure 6-4 Tivoli Enterprise Portal Server log information file

6.3.3 Enabling tracing

When investigating a problem with a particular component, there are two ways to turn up the tracing level for diagnosis:

- Manage Tivoli Enterprise Monitoring Services
- IBM Tivoli Monitoring Service Console

Manage Tivoli Enterprise Monitoring Services

To increase tracing through the Manage Tivoli Enterprise Monitoring Services user interface, perform the following steps:

1. Right-click the desired component and select **Advanced** → **Edit Trace Params** (Figure 6-5).

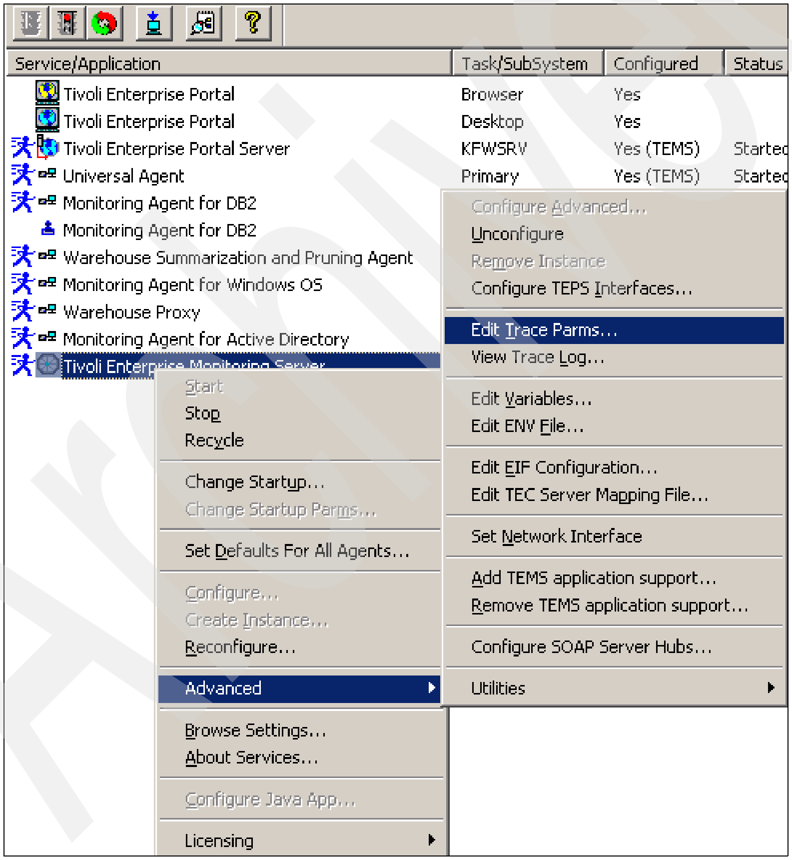


Figure 6-5 Tracing interface example: Tivoli Enterprise Monitoring Server

2. This displays a menu in which you can modify the trace settings (Figure 6-6). This panel is used to specify the number and size of the log files and to select a trace level from the pull-down list. Tracing changes that are set here do not take effect until the component is restarted.

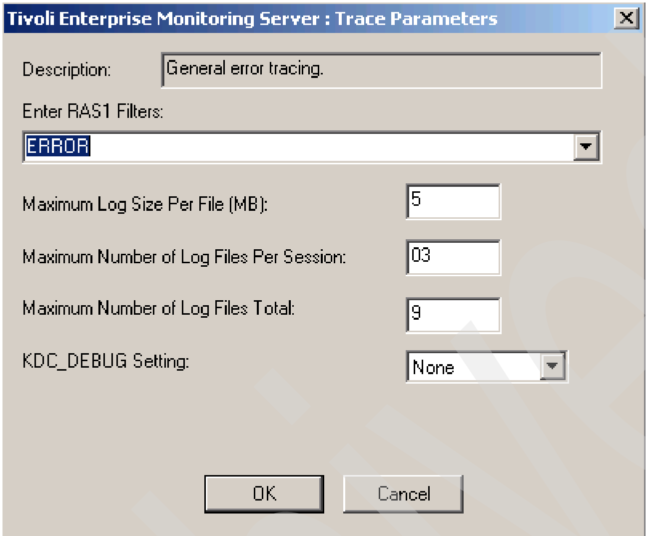


Figure 6-6 Trace Parameters for Tivoli Enterprise Monitoring Server

IBM Tivoli Monitoring Service Console

The IBM Tivoli Monitoring Service Console enables remote product diagnostics and configuration using an industry-standard browser.

Perform the following steps to connect to the IBM Tivoli Monitoring Service Index using a browser:

1. Use a browser to access the following URL (Figure 6-7 on page 359):

   `http://`*`systemname`*`:1920`

   Where *systemname* is the host name of the monitoring and portal server.

2. This is the IBM Tivoli Monitoring Service Console login dialog box. In secure environments, you require a valid user ID and password to proceed. In the Tivoli internal network, select **OK**, leaving the user name and password boxes empty (blank).

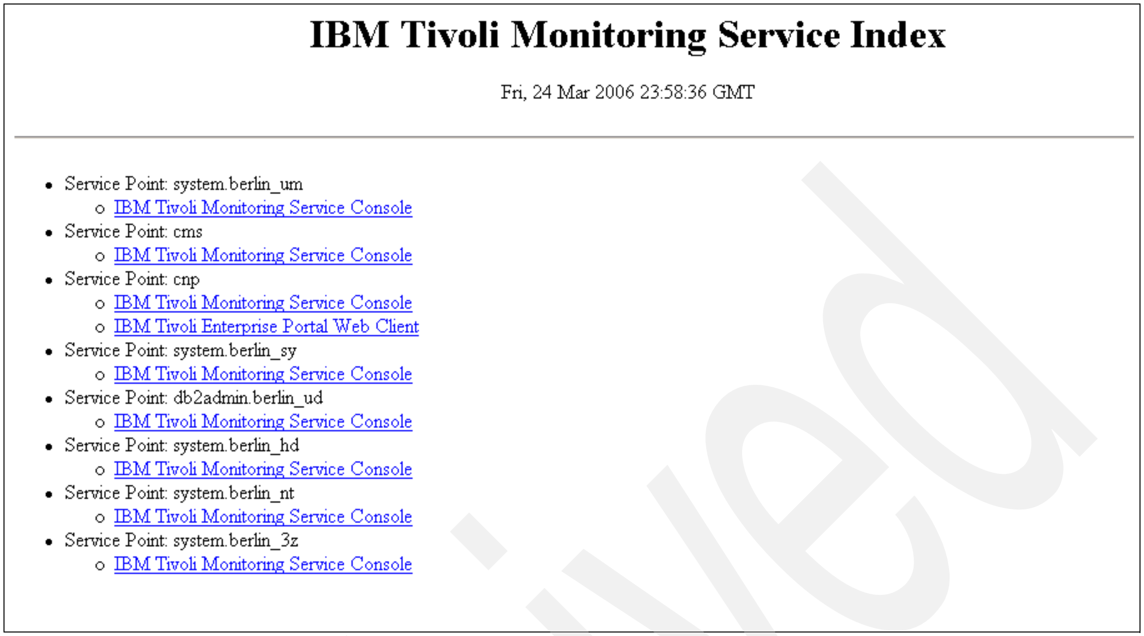


Figure 6-7 IBM Tivoli Monitoring Service Index

3. Each product installed on the system has a corresponding Service Console, as shown in Table 6-3.

Table 6-3 Service Console components

Console	Component
SY	Summarization and Pruning
UM	Universal Agent
HD	Warehouse Proxy agent
NT	Windows OS agent
CNP	Tivoli Enterprise Portal Server
CMS	Tivoli Enterprise Monitoring Server
3Z	Active Directory agent

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 the settings. You can enter trace parameters in the text box at the bottom of the window (Figure 6-8).

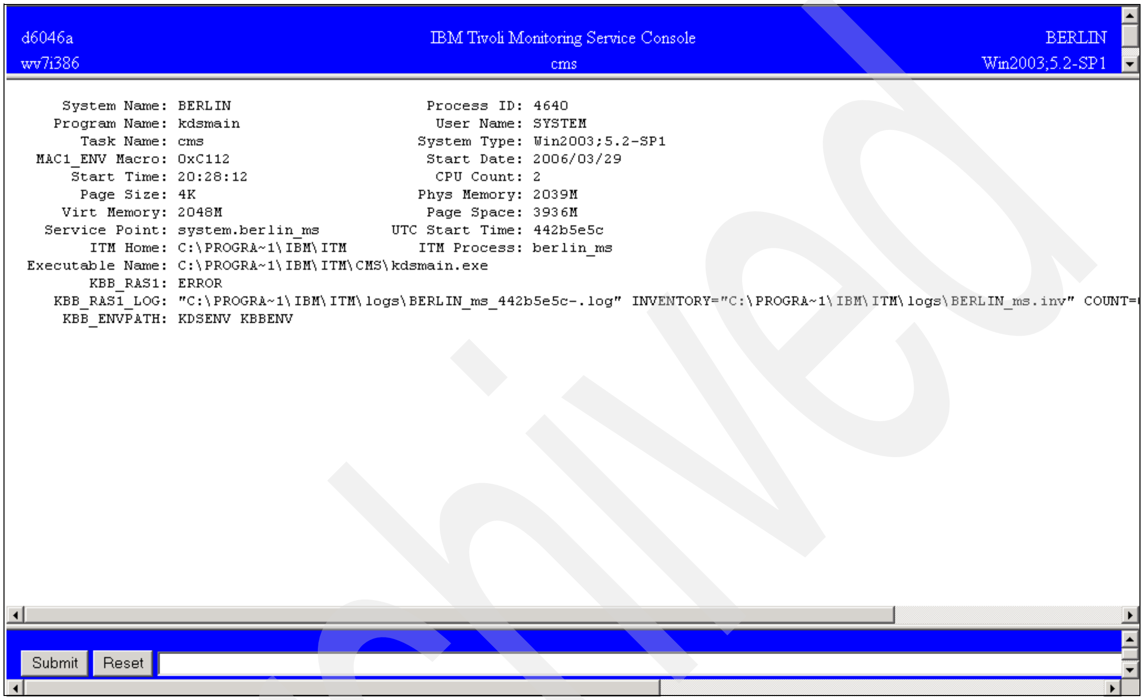


Figure 6-8 IBM Tivoli Monitoring Service Console

The advantage of setting tracing in this manner is that it takes effect dynamically. For more details about using and blocking this tool, see *IBM Tivoli Monitoring V6.1 Problem Determination Guide*, GC32-9458.

4. After modifying the trace settings, recycle the corresponding component for the change to take effect. The log file for the component shows the current trace level in the header, as shown in Example 6-1.

Example 6-1 Header from the Tivoli Enterprise Monitoring Server log file

```
!442B5E5C.0000!========================>  IBM Tivoli RAS1 Service Log
<========================
+442B5E5C.0000      System Name: BERLIN                      Process ID: 4640
+442B5E5C.0000     Program Name: kdsmain                      User Name: SYSTEM
+442B5E5C.0000        Task Name: cms                        System Type:
Win2003;5.2-SP1
+442B5E5C.0000   MAC1_ENV Macro: 0xC112                      Start Date:
2006/03/29
+442B5E5C.0000       Start Time: 20:28:12                     CPU Count: 2
+442B5E5C.0000        Page Size: 4K                       Phys Memory: 2039M
+442B5E5C.0000      Virt Memory: 2048M                      Page Space: 3936M
+442B5E5C.0000    Service Point: system.berlin_ms       UTC Start Time:
442b5e5c
+442B5E5C.0000         ITM Home: C:\PROGRA~1\IBM\ITM        ITM Process:
berlin_ms
+442B5E5C.0000  Executable Name: C:\PROGRA~1\IBM\ITM\CMS\kdsmain.exe
+442B5E5C.0000         KBB_RAS1: ERROR
+442B5E5C.0000     KBB_RAS1_LOG:
"C:\PROGRA~1\IBM\ITM\logs\BERLIN_ms_442b5e5c-.log"
INVENTORY="C:\PROGRA~1\IBM\ITM\logs\BERLIN_ms.inv" COUNT=03 LIMIT=5 PRESERVE=1
MAXFILES=9
+442B5E5C.0000      KBB_ENVPATH: KDSENV KBBENV
+442B5E5C.0000
===============================================================================
```

6.3.4 Using the trace logs

Because several trace logs are likely to be on a given system, it is advisable to know when the error occurred so that you can access the correct log file. After collecting the correct log file, you can view it using any text editor or word processing program. If you use one of these programs to view the log, the hexadecimal time stamp will not be converted. However, if you use the TMS Log Viewer to view the logs, this time stamp will be converted to human-readable format. You can access the TMS Log Viewer through the Manage Tivoli Enterprise Monitoring Services as shown in the Tivoli Enterprise Portal Server (Figure 6-11 on page 363).

To open the TMS Log Viewer for Tivoli Enterprise Portal Server, follow these steps:

1. Right-click **Tivoli Enterprise Portal Server** → **Advanced** → **View Trace Log** (Figure 6-9).

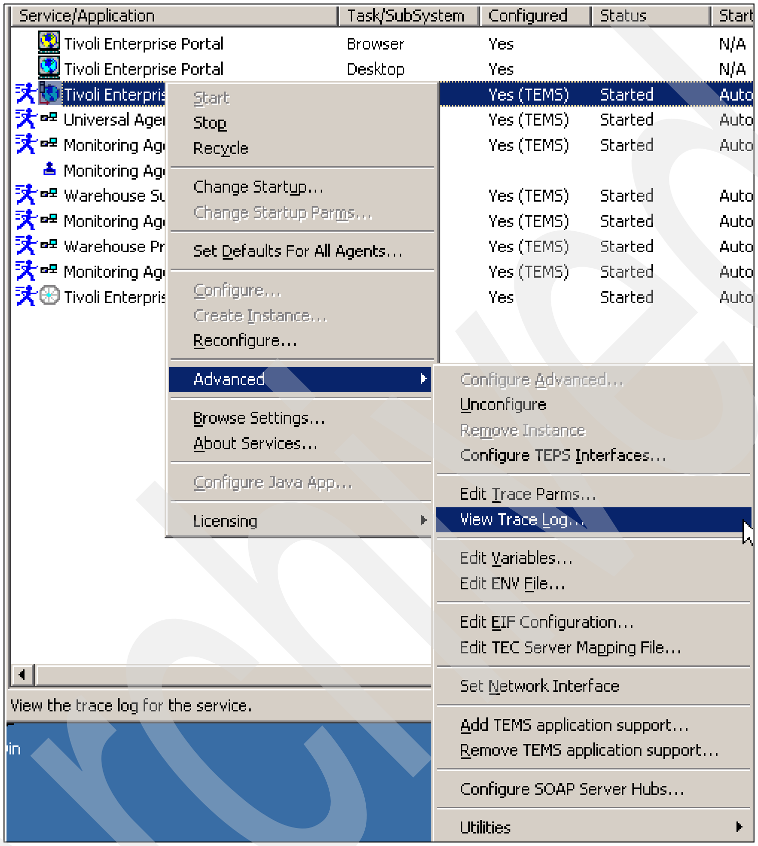


Figure 6-9 Example of the TMS Log Viewer: Tivoli Enterprise Portal Server

2. Select the log file of your choice (Figure 6-10). Click **OK**.

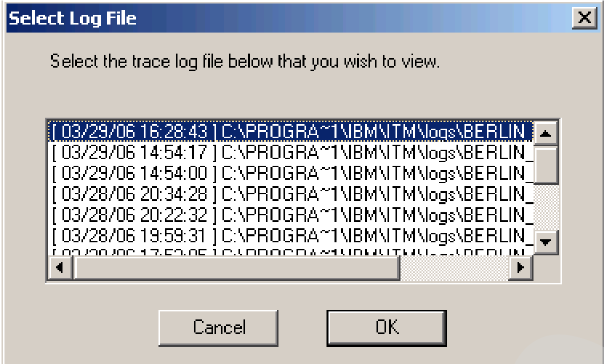


Figure 6-10 Available Tivoli Enterprise Portal Server logs

This opens the TMS Log Viewer, as shown in Figure 6-11.

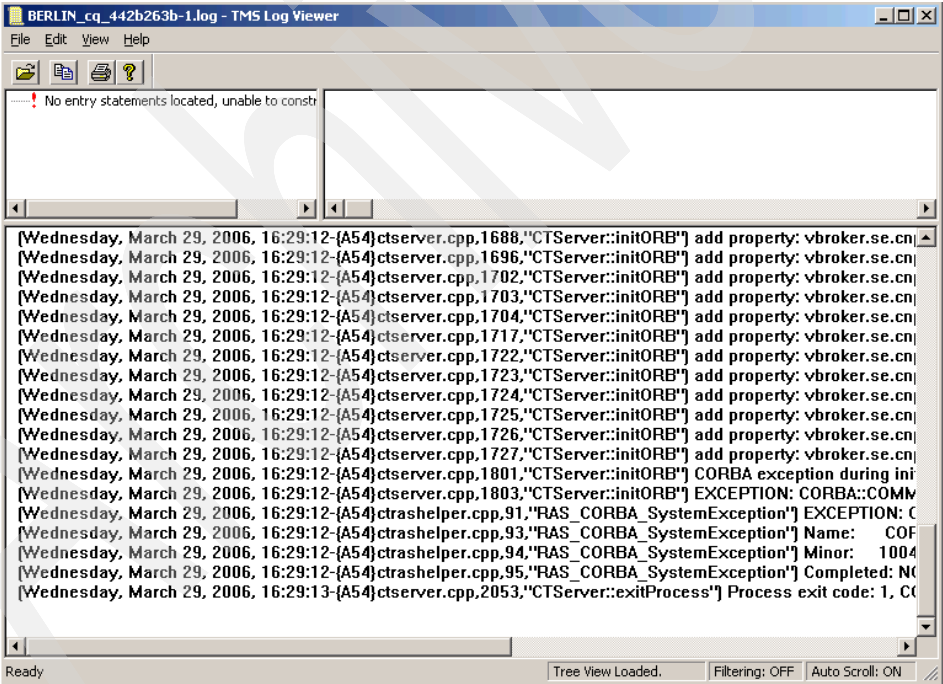


Figure 6-11 The TMS Log Viewer

Note: You can use the TMS Log Viewer for other components (Tivoli Enterprise Monitoring Server, Tivoli Enterprise Monitoring Agents, Warehouse Proxy agent, and Warehouse Summarization and Pruning agent) through the Tivoli Enterprise Monitoring Services interface.

Knowing the type of issue that is 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 Tivoli Enterprise Portal Server logon failure, you can look at the Tivoli Enterprise Portal Server log for the user ID entered when the failure occurred. (Note the response invalid user ID in Example 6-2.)

Example 6-2 Example trace output

```
(442496EC.0000-1514:ctauthorizationevaluator_i.cpp,727,"CTAuthorization::Evalua
tor_i::executeQuery") Invalid Userid <test>
```

The trace level in Example 6-2 was the default of ERROR and the user did not exist in the Tivoli Enterprise Portal Server database.

In Example 6-3, the trace level is increased and the user does exist in the Tivoli Enterprise Portal Server database.

Example 6-3 Example trace output

```
(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
1
```

If the issue is with the **tacmd viewSit** command and the trace level is increased, looking in the kuiras1.log file shows an output similar to Example 6-4.

Example 6-4 Example trace output

```
(442B737F.0035-94C:kuiviewsit.cpp,103,"viewsit") Compiled: Oct 25 2005 21:39:28
+442B737F.0035      Level=1.2, Comp=*
(442B737F.0036-94C:kuiviewsit.cpp,103,"viewsit") Active RAS1 Classes: EVERYT
EVERYE EVERYU
(442B737F.0037-94C:kuiviewsit.cpp,103,"viewsit") Entry
(442B737F.0038-94C:kuiviewsit.cpp,185,"viewsit") SQL QUERY FOR SITDESC SELECT
ADVISE,AFFINITIES,ALERTLIST,AUTOSOPT,AUTOSTART,CMD,DESTNODE,HUB,LO
CFLAG,LSTCCSID,LSTDATE,LSTRELEASE,LSTUSRPRF,NOTIFYARGS,NOTIFYOPTS
,OBJECTLOCK,PDT,PRNAMES,QIBSCOPE,REEV_DAYS,REEV_TIME,REFLEXOK,SENDMSGQ,SITINFO,
```

```
SITNAME,SOURCE,TEXT FROM O4SRV.TSITDESC WHERE SITNAME
='NT_Percent_Processor_Time_Low'
(442B737F.0039-94C:kuiviewsit.cpp,432,"viewsit") Exit: 0x0
(442B737F.003A-94C:kuitacmdmain.cpp,200,"main") Password exists
(442B737F.003B-94C:kuiviewsit.cpp,103,"viewsit") Entry
(442B737F.003C-94C:kuiviewsit.cpp,237,"viewsit") SQL QUERY FOR Distribution
SELECT OBJNAME,NODEL FROM O4SRV.TOBJACCL WHERE
OBJNAME='NT_Percent_Processor_Time_Low' AND SYSTEM.PARMA("QIBNODE",
"QOMEGAVIEW", 32)
(442B737F.003D-94C:kuiviewsit.cpp,432,"viewsit") Exit: 0x0
(442B737F.003E-94C:kuitacmdmain.cpp,200,"main") Password exists
(442B737F.003F-94C:kuitacmdmain.cpp,211,"main") Want to send soap request
(442B737F.0040-94C:RAS1,400,"CTBLD")
+442B737F.0040    Component: kdh
+442B737F.0040       Driver: d6046a/3612569.4
+442B737F.0040    Timestamp: Feb 15 2006 20:53:51
+442B737F.0040       Target: wv7i386
```

Note that when the trace level is increased, the level of details in the log is greatly increased. Keep this in mind when using these logs, and you see that some entries that appear to be errors 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 that is addressed and to have a knowledge of the system to know what you can ignore.

Additionally, when troubleshooting a server issue, it is important to understand the configuration of the servers. Some of the things to consider include:

- Security: Enabled/disabled?
- Protocol in use?
- Firewall in use?
- Is it configured to use Internet Protocol (IP) or host name?
- What port number is used?

6.4 Using the product documentation

When troubleshooting problems, the product documentation is helpful. *IBM Tivoli Monitoring V6.1 Problem Determination Guide*, GC32-9458, covers many topics related to troubleshooting the IBM Tivoli Monitoring Express V6.1 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 guides include problem determination appendixes with information that is specific to the agent.

6.5 Sample problem scenarios

This section provides some samples of the issues encountered in the product's development and testing, as well as the IBM Tivoli Monitoring Express V6.1 beta program. We include the following samples:

- Failure to log on to the Tivoli Enterprise Portal client.
- Failure of the command line to list situations (**`tacmd`**).
- Tivoli Enterprise Portal desktop shows an agent incorrectly but the Tivoli Enterprise Portal browser works fine.

6.5.1 Logging on to the Tivoli Enterprise Portal client fails

A user attempts to log on to the Tivoli Enterprise Portal through 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 enters the user ID and password, the result is a logon failure as shown in Figure 6-12.

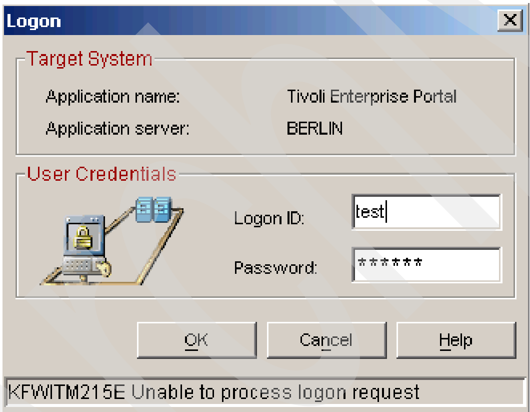


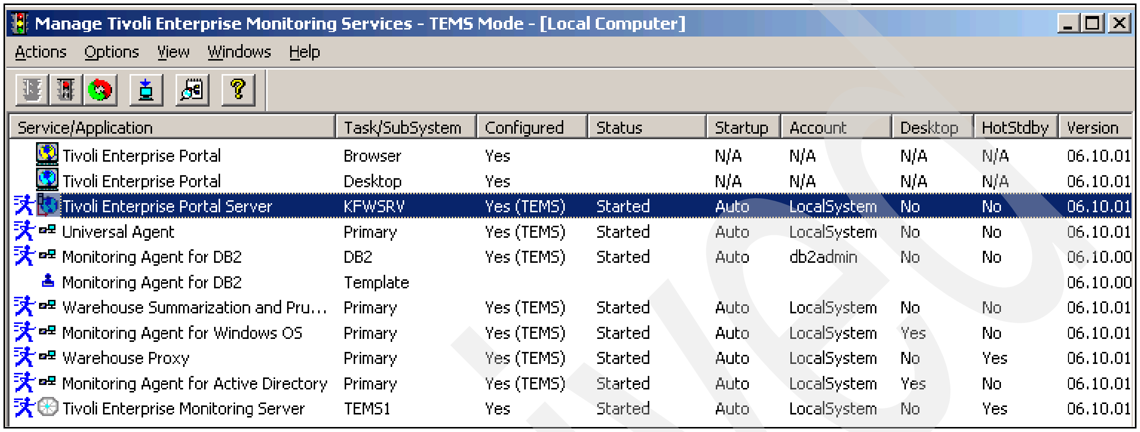
Figure 6-12 Logon failure message

You can isolate and correct this issue by verifying the following questions to troubleshoot it:

- Is the Tivoli Enterprise Portal Server running?
- Is the Tivoli Enterprise Portal Server database up?
- Is the user defined to the Tivoli Enterprise Portal Server?
- Is the password entered correctly?
- Is security enabled on the Tivoli Enterprise Monitoring Server?
- Is the Tivoli Enterprise Portal Server connecting to the Tivoli Enterprise Monitoring Server?

The location of the Tivoli Enterprise Portal Server determines the method used to verify whether the server is running. Perform the following steps:

1. Open Manage Tivoli Enterprise Monitoring Services and check to see whether the Tivoli Enterprise Portal Server is started, as shown in Figure 6-13.



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
Tivoli Enterprise Portal	Browser	Yes		N/A	N/A	N/A	N/A	06.10.01
Tivoli Enterprise Portal	Desktop	Yes		N/A	N/A	N/A	N/A	06.10.01
Tivoli Enterprise Portal Server	KFWSRV	Yes (TEMS)	Started	Auto	LocalSystem	No	No	06.10.01
Universal Agent	Primary	Yes (TEMS)	Started	Auto	LocalSystem	No	No	06.10.01
Monitoring Agent for DB2	DB2	Yes (TEMS)	Started	Auto	db2admin	No	No	06.10.00
Monitoring Agent for DB2	Template							06.10.00
Warehouse Summarization and Pru...	Primary	Yes (TEMS)	Started	Auto	LocalSystem	No	No	06.10.01
Monitoring Agent for Windows OS	Primary	Yes (TEMS)	Started	Auto	LocalSystem	Yes	No	06.10.01
Warehouse Proxy	Primary	Yes (TEMS)	Started	Auto	LocalSystem	No	Yes	06.10.01
Monitoring Agent for Active Directory	Primary	Yes (TEMS)	Started	Auto	LocalSystem	Yes	No	06.10.01
Tivoli Enterprise Monitoring Server	TEMS1	Yes	Started	Auto	LocalSystem	No	Yes	06.10.01

Figure 6-13 Manage Tivoli Monitoring Services menu

2. From a command window on the Tivoli Enterprise Portal Server system, type the following command:

```
C:\PROGRA~1\IBM\ITM\InstallITM\kincinfo -r
```

Example 6-5 shows the output.

Example 6-5 kincinfo -r output

```
C:\IBM\ITM\Install>kincinfo -r
*********** Tue Mar 28 11:10:04 Pacific Standard Time 2006 *****************
User      : Administrator Group    : NA
Host Name : BERLIN       Installer: Ver: NOVALUE
CandleHome: C:\PROGRA~1\IBM\ITM
*********************************************************************
Host          Prod  PID      Owner                 Start     Status Task
BERLIN        MS       3368 NT AUTHORITY\SYSTEM    10:53:18  ..Running TEMS1
BERLIN        FW        624 NT AUTHORITY\SYSTEM    10:56:38  ..Running KFWSRV
BERLIN        3Z       5336 NT AUTHORITY\SYSTEM    11:07:40  ..Running Primary
BERLIN        HD       6048 NT AUTHORITY\SYSTEM    12:26:34  ..Running Primary
BERLIN        NT       5788 NT AUTHORITY\SYSTEM    15:38:34  ..Running Primary
BERLIN        SY       5172 NT AUTHORITY\SYSTEM    12:26:41  ..Running Primary
BERLIN        UD          0 NONE                    0:00:00  ..Not Running
BERLIN        UD          0 NONE                    0:00:00  ..Not Running
```

```
BERLIN       UM       4816 NT AUTHORITY\SYSTEM   11:43:37  ..Running Primary
BERLIN       IN       6004 BERLINO\Administrator  12:13:04  ..Running

FW code corresponds to the TEPS process on Windows.
```

When logging on to the Tivoli Enterprise Portal Server, the first check is made to the Tivoli Enterprise Portal Server table that defines the user to the system: TEPS.KFWUSER. This check can be seen in the Tivoli Enterprise Portal Server logs shown in Example 6-6.

Example 6-6 Two examples from the Tivoli Enterprise Portal Server logon user

```
User not defined:

(442496EC.0000-1514:ctauthorizationevaluator_i.cpp,727,"CTAuthorization::Evalua
tor_i::executeQuery") Invalid Userid <test>
(442496EC.0001-1514:ctdatabusmanager_i.cpp,820,"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 = 'sysadmin')
(43682628.0015-B7C:ctsqlaccessodbc.cpp,1007,"CTSQLEvaluatorODBC_i::AccessElemen
t::pullSequenceWithTimeout") TEPS2(69): Rows returned: 1
```

The password is not seen in the logs. This is the password of the user on the Tivoli Enterprise Monitoring Server system. If the user exists and the password is questioned, the user should try again with the known password or attempt to reset the password on the Tivoli Enterprise Monitoring Server operating system. The password is not stored in the Tivoli Enterprise Portal Server or the Tivoli Enterprise Monitoring Server, and the user is validated on the Tivoli Enterprise Monitoring Server during the logon process to the operating system.

You can check whether the security is enabled on the Tivoli Enterprise Monitoring Server system in the logs and by looking at the monitoring server (Example 6-7).

Example 6-7 Entry in the Tivoli Enterprise Monitoring Server log

```
+44299095.0022    Timestamp: Feb 15 2006 21:23:28
+44299095.0022       Target: wv7i386
(44299095.0023-6BC:kbbssge.c,52,"BSS1_GetEnv") CMS_VALIDATE="YES"
```

If the Tivoli Enterprise Portal Server and the Tivoli Enterprise Monitoring Server are not communicating, the error message is different (for example, `KFWITM001W Unable to connect to Tivoli Enterprise Portal Server`). If the Tivoli Enterprise Portal Server shows that it is up, a likely cause is that the Tivoli Enterprise Monitoring Server is down or not responding to the Tivoli Enterprise Portal Server logon request.

The following solutions to this situation depend on the results of the troubleshooting:

- Start the Tivoli Enterprise Portal Server.
- 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 Tivoli Enterprise Monitoring Server.

6.5.2 Command line fails to list situations (tacmd)

When using the **tacmd** command to view a situation, unexpected results are produced. The **tacmd** command logs to ITM_InstallDir\bin\kuiras1.log on Windows and ITM_InstallDir/logs/kuiras1.log on UNIX. With increased trace settings, the situation can be seen in the logs. If there is an issue, it is not seen in the logs.

In this log, the trace is set to KBB_RAS1=ERROR(UNIT:KUI ALL). Example 6-8 shows the details of the situation.

Example 6-8 Situation in kuiras1.log

```
(442B737F.0037-94C:kuiviewsit.cpp,103,"viewsit") Entry
(442B737F.0038-94C: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 ='NT_Percent_Processor_Time_Low'
(442B737F.0039-94C:kuiviewsit.cpp,432,"viewsit") Exit: 0x0
(442B737F.003A-94C:kuitacmdmain.cpp,200,"main") Password exists
(442B737F.003B-94C:kuiviewsit.cpp,103,"viewsit") Entry
(442B737F.003C-94C:kuiviewsit.cpp,237,"viewsit") SQL QUERY FOR Distribution
SELECT OBJNAME,NODEL FROM O4SRV.TOBJACCL WHERE
OBJNAME='NT_Percent_Processor_Time_Low' AND SYSTEM.PARMA("QIBNODE",
"QOMEGAVIEW", 32)
(442B737F.003D-94C:kuiviewsit.cpp,432,"viewsit") Exit: 0x0
.....
(442B7381.0006-94C:kuiviewsit.cpp,103,"viewsit") Entry
(442B7381.0007-94C:kuiviewsit.cpp,448,"processResponse") Active RAS1 Classes:
EVERYT EVERYE EVERYU
```

```
(442B7381.0008-94C:kuiviewsit.cpp,448,"processResponse") Entry
(442B7381.0009-94C:kuiviewsit.cpp,570,"processResponse") Number of Rows in
TOBJACCL for sitname  is 0
(442B7381.000A-94C:kuiviewsit.cpp,572,"processResponse") Number of Rows in
TSITDESC for sitname  is 1
(442B7381.000B-94C:kuiviewsit.cpp,649,"processResponse") Distribution Info
(442B7381.000C-94C:kuiviewsit.cpp,661,"processResponse") SITNAME Info
NT_Percent_Processor_Time_Low
(442B7381.000D-94C:kuiviewsit.cpp,673,"processResponse") TEXT Knt:KNT1344
(442B7381.000E-94C:kuiviewsit.cpp,685,"processResponse") AFFINITIES
(442B7381.000F-94C:kuiviewsit.cpp,696,"processResponse") PDT *IF *VALUE
NT_Processor.%_Processor_Time *LE 10
(442B7381.0010-94C:kuiviewsit.cpp,709,"processResponse") SAMPLING INTERVAL 0
001500
(442B7381.0011-94C:kuiviewsit.cpp,722,"processResponse") AUTOSTART *NO
(442B7381.0012-94C:kuiviewsit.cpp,733,"processResponse") ADVISE
ADVICE("knt:"+$ISITSTSH.SITNAME$);
(442B7381.0013-94C:kuiviewsit.cpp,744,"processResponse") CMD *NONE
(442B7381.0014-94C:kuiviewsit.cpp,755,"processResponse") AUTOSOPT NN
(442B7381.0015-94C:kuiviewsit.cpp,770,"processResponse") ALERTLIST *NO
(442B7381.0016-94C:kuiviewsit.cpp,792,"processResponse") QIBSCOPE E
(442B7381.0017-94C:kuiviewsit.cpp,803,"processResponse") SENDMESGQ *NONE
(442B7381.0018-94C:kuiviewsit.cpp,846,"processResponse") LSTCCSID
0970901010101000
(442B7381.0019-94C:kuiviewsit.cpp,857,"processResponse") LSTRELEASE V301
(442B7381.001A-94C:kuiviewsit.cpp,868,"processResponse") LSTUSRPRF CCC_V301
(442B7381.001B-94C:kuiviewsit.cpp,934,"processResponse") SITINFO
SEV=Warning;ATOM=NTPROCSSR.INSTCNAME
```

If **tacmd viewSit -s *situation_name*** does not show a situation, check to ensure that the situation exists using the **tacmd listSit** command and that the situation is defined on the hub Tivoli Enterprise Monitoring Server. If a situation is defined on a remote using the **tacmd createSit** command, 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 Tivoli Enterprise Monitoring Server.

6.5.3 Tivoli Enterprise Portal desktop shows an agent incorrectly

The Tivoli Enterprise Portal desktop client shows an agent with incorrect attribute group label names, as shown in Figure 6-14 on page 371. The Tivoli Enterprise Portal browser client shows the labels correctly, so what is wrong and how can you correct this?

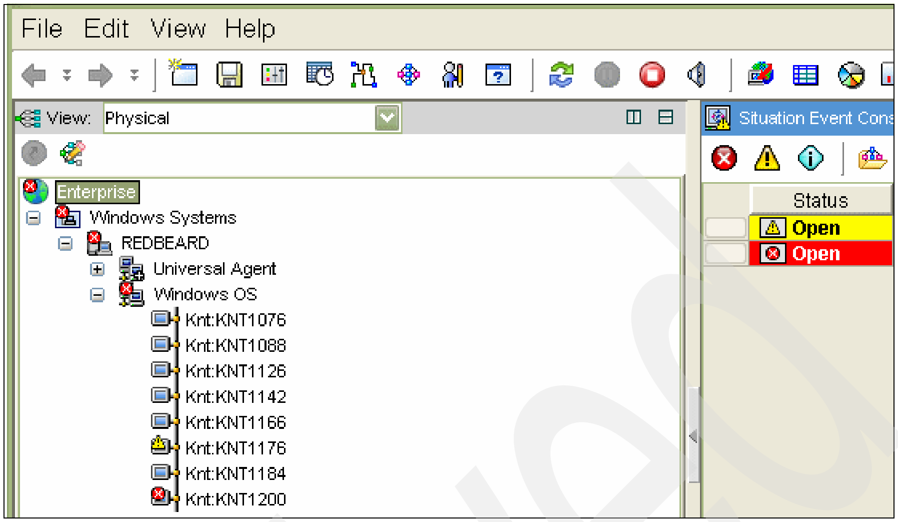


Figure 6-14 Tivoli Enterprise Portal Desktop client: Incorrect agent attribute group labels

This can point to one of the following two issues:

- The application seeding is not complete.
- There is an incorrect class path for the Tivoli Enterprise Portal desktop.

In this case, a review of the CMSseed.log file shows the output illustrated in Example 6-9.

Example 6-9 CMSseed.log

```
Addition of application support for component: knt
completed with rc: 0
C:\Program Files\IBM\ITM\CNPS\sqllib\knt.sql
Output from the operation was written to log file:
C:\Program Files\IBM\ITM\CNPS\logs\seedknt.log
```

Therefore, the Windows monitoring agent support appears to be loaded correctly in the Tivoli Enterprise Monitoring Server, and these attribute group labels should show up correctly. In addition, the Tivoli Enterprise Portal browser shows the correct labels pointing to a problem with the desktop client.

An investigation of the desktop client log reveals the cause of the problem, as shown in Example 6-10.

Example 6-10 Tivoli Enterprise Portal 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;k3z_resources.jar;ka4_resour
ces.jar;klz_resources.jar;kul_resources.jar;kum_resources.jar;kux_resources.jar
;kud_resources.jar;.
......
(4429e873.2634e240-(null)main:KfwBundle,0,"KfwBundle.getBundle()") Version:
1.151.2.2
```

> **Note:** The kcjras1.log file (desktop client) or the plugin1.4.2.trace file contains the same initial environmental information related to the client's desktop environment. The beginning of the log contains all the system properties that Java knows of on that system along with some properties that Tivoli has created and added to the property set.

The java.class.path statement shows that there is no knt_resources.jar file listed. This is the JAR file used by the Tivoli Enterprise Portal desktop to load the Windows agent views. This did not cause the knt package to fail in initializing. The solution is to modify the cnp.bat file, which is used by the Tivoli Enterprise Portal desktop client on Windows, and add the knt_resources.jar. It is also a good idea to verify that the knt_resources.jar file exists on the Tivoli Enterprise Portal desktop client as well. Example 6-11 shows the updated cnp.bat class path statement for the issue.

Example 6-11 Updated cnp.bat class path 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;k3z_resources.jar;
ka4_resources.jar;klz_resources.jar;knt_resources.jar;kul_resources.jar;kum_res
ources.jar;kux_resources.jar;kud_resources.jar
```

> **Important:** The Tivoli Enterprise Portal client has dynamic logging. Restarting the processes before collecting the logs will rewrite the log and any previous error messages can be lost.

6.6 Common installation problems in IBM Tivoli Monitoring Express V6.1

This section lists the most common installation problems that you can incur while installing IBM Tivoli Monitoring Express V6.1. For more comprehensive troubleshooting and problem determination, refer to *IBM Tivoli Monitoring V6.1 Problem Determination Guide*, GC32-9458, at the following Web site:

http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.itm.doc/PDG_ITM6102.htm#wq2

Table 6-4 describes some of the common errors.

Table 6-4 Common errors

Error	Description and resolution
InstallShield displays the Error `1607: Unable to install InstallShield Scripting Runtime` during installation on Windows from a network-mounted drive.	When running setup.exe on Windows from a network-mounted drive, the following error occurs: InstallShield: `1607: Unable to install InstallShield Scripting Runtime`. This is an InstallShield limitation. You cannot install the product from the specified network drive. Try installing from another network drive. Install the product from a local drive if you continue to receive the error.

When running setup.exe, an Unknown Publisher Error message is displayed.	If you run setup.exe from a Universal Naming Convention (UNC) path on Windows, you receive the following message:
	`File Download - Security Warning`
	`The Publisher could not be verified.`
	`Are you sure you want to run this software?`
	Selecting Cancel closes the window and the installation cannot complete. To install the software without this problem, map the path to a network drive and run the setup.exe file from an MS-DOS prompt.

Installation on a Windows Server 2003 fails with error number 0x80040707.	An unhandled exception with error number 0x80040707 can occur while installing IBM Tivoli Monitoring Express V6.1 on a Windows Server 2003. This error can occur for the following reasons: ▶ Windows Service Pack 1 (or later) is not installed. ▶ Windows Installer 3.1 (KB893803) is not installed. KB893803 is included in SP1. You must install Windows Service Pack 1 or later or the KB893803 individual update. You can download the update from the Web site: http://www.windowsupdate.com

You can enable the Launchpad tracing by opening an MS-DOS window and set LaunchPadLogFilter=SEWT. Then, invoke launchpad.exe from the same window. This creates a pane at the bottom of the Launchpad frame that contains the trace output.

Security warnings during an Express Launchpad installation

If you run the Express Launchpad installation from a remote computer, a security warning pop-up window might appear. In this case, you can perform one of the the following options:

- Run the installation from a local CD media.
- Copy all IBM Tivoli Monitoring Express disk images to the local machine and install it.

Checking the warehouse data

After the first full hour, you should start seeing some activity in the Tivoli Data Warehouse tables. You can check this from the DB2 Control Center.

1. To open the DB2 Control Center, select **Start** → **Programs** → **IBM DB2** → **General Administration Tools** → **Control Center**.
2. Expand the **All Databases** tree and then the **WAREHOUS** database.

3. Click **Tables** to see all the tables that exist in the database (Figure 6-15).

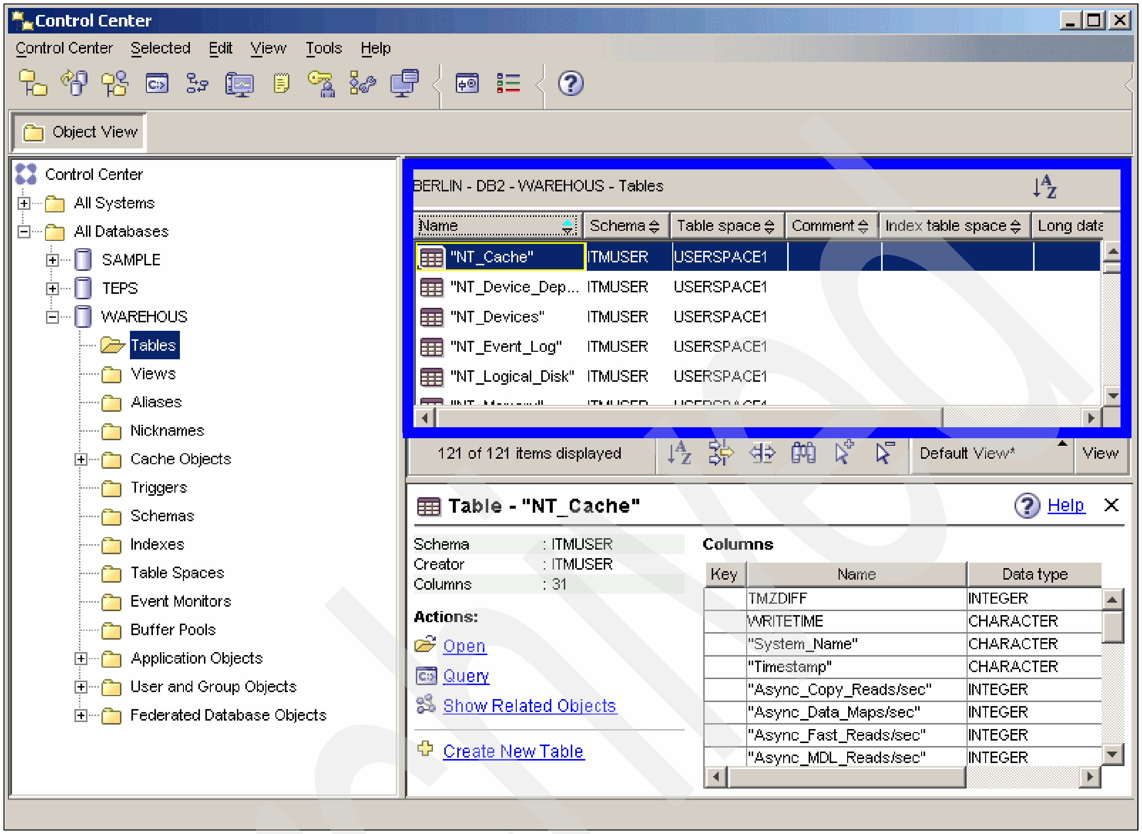


Figure 6-15 Warehouse historical database

You can click the **Schema** header line to sort the schemas alphabetically. This way, all the tables owned by the ITMUser will be at the top.You should recognize the names of the attribute groups as table names in the database, as shown in Figure 6-15.

If you do not have these tables in your database at the top of the next full hour, check to see whether there are any errors in the Warehouse Proxy agent log file (Figure 6-18 on page 377). If there are any connection errors in this log file, you probably made a mistake with the user ID and password combinations when you configured the Warehouse Proxy ODBC settings.

To open the Warehouse Proxy agent log file, perform the following steps:

1. Right-click **Warehouse Proxy** and select **Advanced** → **View Trace Log** (Figure 6-16).

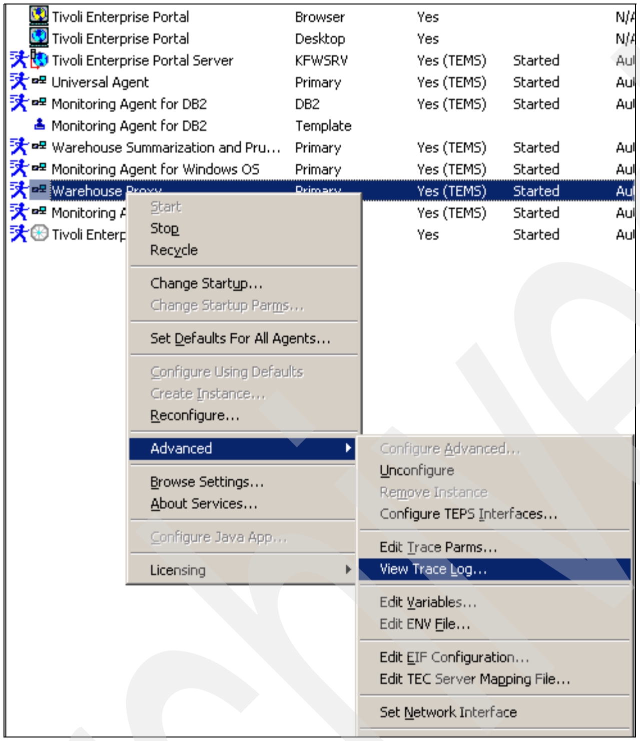


Figure 6-16 Warehouse Proxy Trace Log

2. Select the log file of your choice (Figure 6-17). Click **OK**.

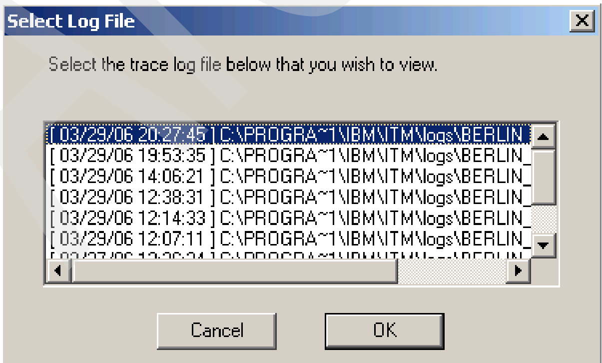


Figure 6-17 Available Warehouse Proxy logs

This opens the Warehouse Proxy Log Viewer, as shown in Figure 6-18.

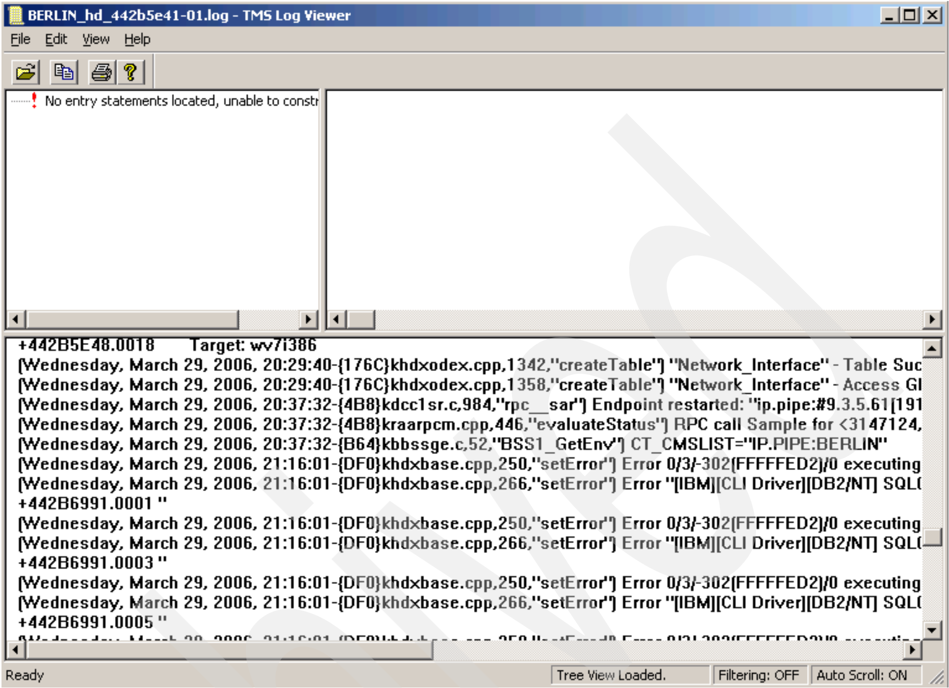
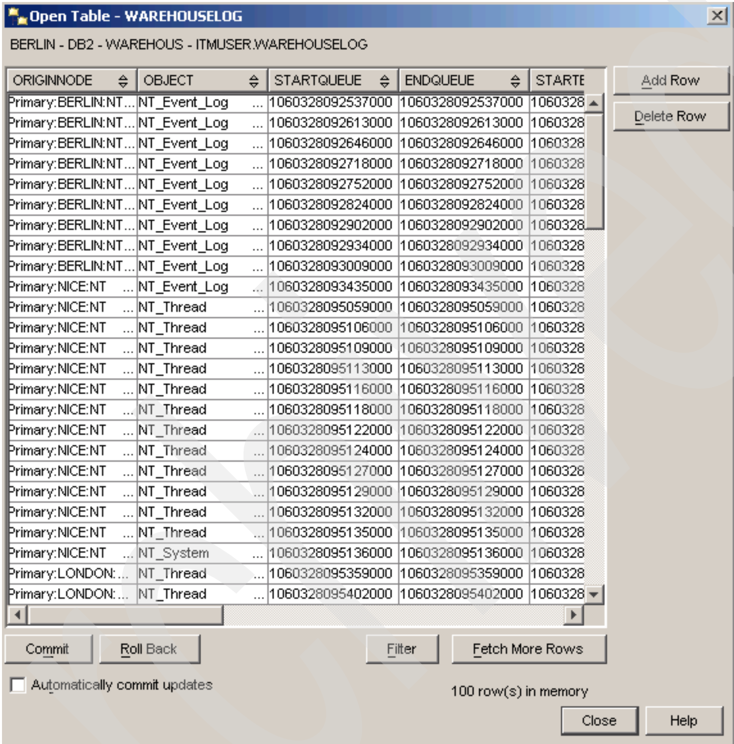


Figure 6-18 Warehouse Proxy Log Viewer

You can now experiment a little more with displaying historical data. However, bear in mind that any data that is less than 24 hours old is pulled from the agent directly. Only if you try to access historical data that is older than 24 hours will the Tivoli Enterprise Portal Server go to the Data Warehouse.

When determining errors about loading data from the Tivoli Enterprise Monitoring Agent to the Tivoli Data Warehouse through the Warehouse Proxy agent, look in the RDBMS database at a table called WAREHOUSELOG, as shown in Figure 6-19. However, this table is not useful in determining Tivoli Enterprise Portal graphical user interface (GUI) display problems related to viewing historical data. Sometimes, timing parameters defined in the Tivoli Enterprise Monitoring Server can affect a user's ability to display historical data.



ORIGINNODE	OBJECT	STARTQUEUE	ENDQUEUE	STARTE
Primary:BERLIN:NT...	NT_Event_Log ...	1060328092537000	1060328092537000	1060328
Primary:BERLIN:NT...	NT_Event_Log ...	1060328092613000	1060328092613000	1060328
Primary:BERLIN:NT...	NT_Event_Log ...	1060328092646000	1060328092646000	1060328
Primary:BERLIN:NT...	NT_Event_Log ...	1060328092718000	1060328092718000	1060328
Primary:BERLIN:NT...	NT_Event_Log ...	1060328092752000	1060328092752000	1060328
Primary:BERLIN:NT...	NT_Event_Log ...	1060328092824000	1060328092824000	1060328
Primary:BERLIN:NT...	NT_Event_Log ...	1060328092902000	1060328092902000	1060328
Primary:BERLIN:NT...	NT_Event_Log ...	1060328092934000	1060328092934000	1060328
Primary:BERLIN:NT...	NT_Event_Log ...	1060328093009000	1060328093009000	1060328
Primary:NICE:NT ...	NT_Event_Log ...	1060328093435000	1060328093435000	1060328
Primary:NICE:NT ...	NT_Thread ...	1060328095059000	1060328095059000	1060328
Primary:NICE:NT ...	NT_Thread ...	1060328095106000	1060328095106000	1060328
Primary:NICE:NT ...	NT_Thread ...	1060328095109000	1060328095109000	1060328
Primary:NICE:NT ...	NT_Thread ...	1060328095113000	1060328095113000	1060328
Primary:NICE:NT ...	NT_Thread ...	1060328095116000	1060328095116000	1060328
Primary:NICE:NT ...	NT_Thread ...	1060328095118000	1060328095118000	1060328
Primary:NICE:NT ...	NT_Thread ...	1060328095122000	1060328095122000	1060328
Primary:NICE:NT ...	NT_Thread ...	1060328095124000	1060328095124000	1060328
Primary:NICE:NT ...	NT_Thread ...	1060328095127000	1060328095127000	1060328
Primary:NICE:NT ...	NT_Thread ...	1060328095129000	1060328095129000	1060328
Primary:NICE:NT ...	NT_Thread ...	1060328095132000	1060328095132000	1060328
Primary:NICE:NT ...	NT_Thread ...	1060328095135000	1060328095135000	1060328
Primary:NICE:NT ...	NT_System ...	1060328095136000	1060328095136000	1060328
Primary:LONDON:...	NT_Thread ...	1060328095359000	1060328095359000	1060328
Primary:LONDON:...	NT_Thread ...	1060328095402000	1060328095402000	1060328

Figure 6-19 WAREHOUSELOG database

Example of common error code: KFWITM220E Request failed during execution

Open the Workspace called **Historical Summarized Capacity** (Figure 6-20).

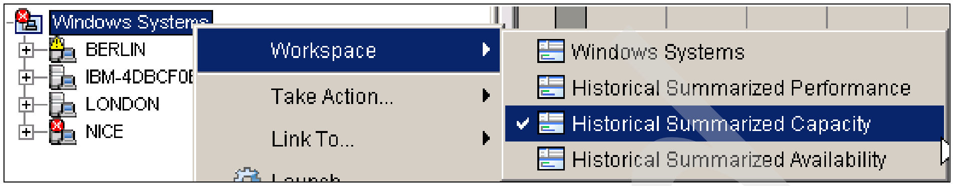


Figure 6-20 Historical Summarized Capacity Workspace

This workspace does not show any data (Figure 6-21).

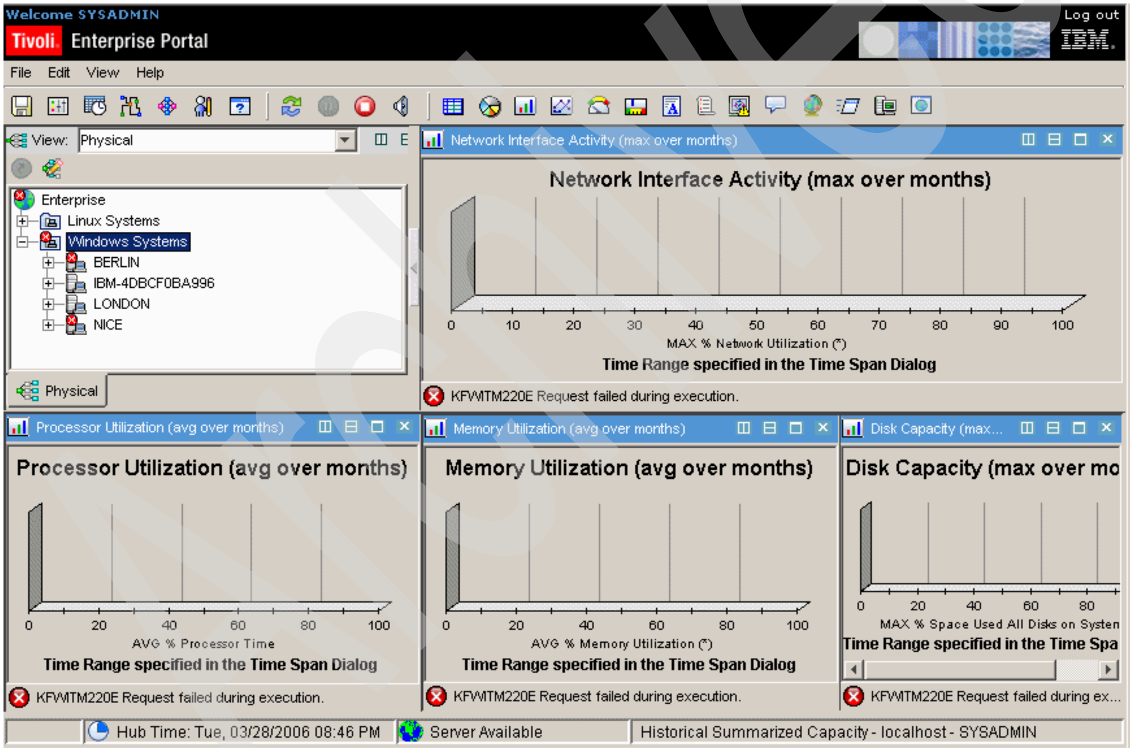


Figure 6-21 Historical Summarized Capacity Workspace

The explanation for this can be:

- Data for historical query is not yet recorded. Check if the Warehouse DataSource is configured correctly in the Tivoli Enterprise Portal Server log file.
- Data Collection interval is not elapsed. Check the Tivoli Enterprise Portal Server log file.

Refer to 4.3, "Historical data collection" on page 191 to configure historical data collection.

6.7 Tivoli Enterprise Portal Server hints

Consider the following Tivoli Enterprise Portal Server hints:

- When logging on to the Tivoli Enterprise Portal Server through a browser, you might see HeapDumps and a JAVACore entries. Make sure that the user's Java environment has the following parameters defined:
  - Xms128m
  - Xmx256m
- The minimum level of JRE for the Tivoli Enterprise Portal Server to run is Java 4.1.2.
- If you are getting an out-of-memory condition, java.lang.OutOfMemoryError, when you are logged on to Tivoli Enterprise Portal, it manifests itself in different ways. For example, in browser mode, the window might be disabled after an hour or two. If you are connecting to multiple portal servers from the same computer, increase the memory by 125 MB for each portal server. Review the Java log plugin142.trace file on your system to confirm the cause. On a Windows system, this is located in C:\Documents and Settings\Administrator\Application Data\IBM\Java\Deployment\log.

Tivoli Enterprise Portal Server trace settings

Note the following Tivoli Enterprise Portal Server trace settings:s

- To gather general problem context for client requests: ERROR (UNIT:ctsql INPUT,ERROR), (UNIT:ctdatabus INPUT,ERROR).
- To see the effect Tivoli Enterprise Monitoring Server events have: ERROR (UNIT:ctsql INPUT,ERROR), (UNIT:ctdatabus INPUT,ERROR), (UNIT:kv4mvmdl INPUT,ERROR).

- Set the trace options for the Tivoli Enterprise Portal Server when you start the portal server. Before you set the trace options for the portal server, determine the trace string. The trace string specifies the trace setting. The log file continues to grow until you either turn off the trace or recycle the portal server:
  a. On the computer where the Tivoli Enterprise Portal Server is installed, select **Start** → **Programs** → **Manage Tivoli Monitoring Services** → **Manage Tivoli Monitoring Services**.
  b. After setting the trace parameters, click **OK** (Figure 6-22). The Tivoli Enterprise Portal Server is restarted with the specified trace set.
  c. If you are instructed to enter a value for KDC_DEBUG Setting, you can also do this from this window. Although you can modify the trace log file name, we recommend that you do not do this unless instructed by the Tivoli Enterprise Portal Level 3 support.

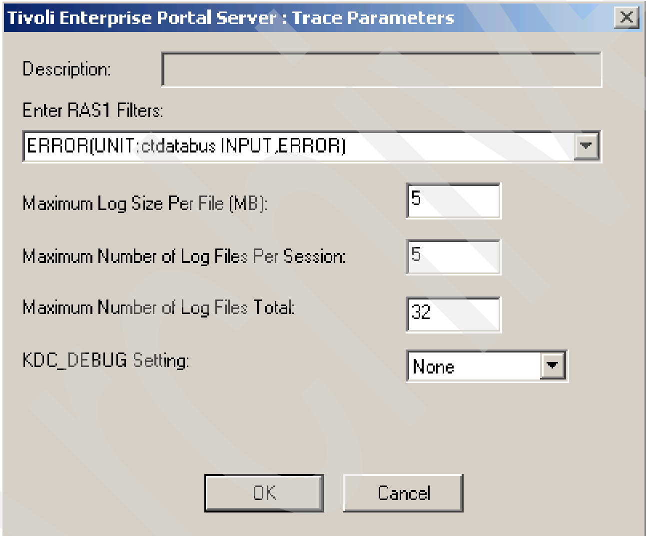


Figure 6-22 Tivoli Enterprise Portal Server: Trace Parameters

6.8 Tivoli Enterprise Monitoring Server hints

The following hints and tips concern the Tivoli Enterprise Monitoring Server:

- Always seed the Tivoli Enterprise Monitoring Server immediately after the installation. Remember to reboot the Tivoli Enterprise Monitoring Server right after seeding. This prevents blank workspaces in the Tivoli Enterprise Portal because the monitoring server does not support the attribute groups.

- Start the Tivoli Enterprise Monitoring Server in order to seed for application support.
- We recommend that you do not enable security validation when installing the hub Tivoli Enterprise Monitoring Server initially. Set up the environment completely before enabling security validation within the installation. Remember that sysadmin is the default administrator user ID.
- You can set the CTIRA_HEARTBEAT variable in the KBBENV file, as shown in Example 6-12, to specify the heartbeat interval among the hub Tivoli Enterprise Monitoring Server. The default interval is 5 minutes.

Example 6-12 Example of setting the heartbeat to 5 minutes

```
KGL_TRC1=ENABLE ERRLOG
KDS_LBREG=YES
KDS_NCS=YES
KDS_CATLGLIB=QA1CDSCA
KDS_RULELIB=QA1CRULD
KDS_START=KDSPRB.KDSOPTSK
CMS_EXTERNALBROKERS=NO
NLS1_LOCALEDIR=^>C:\PROGRA~1\IBM\ITM\CMS\locale
KDH_SERVICEPOINT=cms
KGL_MSG2_UNIVERSAL=YES
CANDLE_HOME=C:\PROGRA~1\IBM\ITM
CTIRA_HEARTBEAT=300
```

6.9 Tivoli Enterprise Portal troubleshooting

To troubleshoot the Tivoli Enterprise Portal component, consider the following points:

- Tivoli Enterprise Portal client logs location:

  Windows: *ITM_InstallDir*\CNP\logs

  In this case, *ITM_InstallDir* is the installation directory of IBM Tivoli Monitoring Express.

  The Tivoli Enterprise Portal client logs contain environmental information such as the version and build level of the Tivoli Enterprise Portal client. The log also contains the host and port of the Tivoli Enterprise Portal Server to which the client is connecting.

- TEP browser client logs:
  – KCJ.LOG (Example 6-13) contains any errors that might be thrown by the Java libraries used in the Tivoli Enterprise Portal client.

Example 6-13 KCJ.LOG

```
Using Java Release: 1.4.2
Java Home = C:\Program Files\IBM\Java142\jre
java version "1.4.2"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.4.2)
Classic VM (build 1.4.2, J2RE 1.4.2 IBM Windows 32 build cn142sr1a-20050209
(JIT enabled: jitc))
C:\Program Files\IBM\ITM\CNP\CNP.BAT
```

  – kcjerror.log (Example 6-14) contains messages, errors, and exceptions that come from the third-party products used in the Tivoli Enterprise Portal client.

Example 6-14 kcjerror.log

```
Tue 03/28/2006
07:13 PM
ICEssl v3_0_4
(c) ICEsoft Technologies, Inc.
ICEhttp v1_6_2
(c) ICEsoft Technologies, Inc.
ICEbrowser v6_1_2
(c) ICEsoft Technologies, Inc.
```

The plugin1.4.2.trace file contains the RAS1 tracing for the Tivoli Enterprise Portal browser client and any Java exceptions.

Setting trace: Tivoli Enterprise Portal client

A log file is created automatically the first time you start the Tivoli Enterprise Portal named ITM_InstallDir\logs\kcjras1.log. This log file contains all of the RAS1 tracing for the Tivoli Enterprise Portal client. Whenever you start a new work session, the log file is purged and rewritten for the current work session. If you want to preserve the log file from the last work session, you must rename it or copy it to another directory before starting the Tivoli Enterprise Portal again. The kcj.log file contains errors generated by the Sun Java™ libraries used in the Tivoli Enterprise Portal client.

Perform the following steps:

1. The Tivoli Enterprise Portal clients have the ability to set RAS1 tracing dynamically. From the Tivoli Enterprise Portal menu, select **File** → **Trace Options** (Figure 6-23).

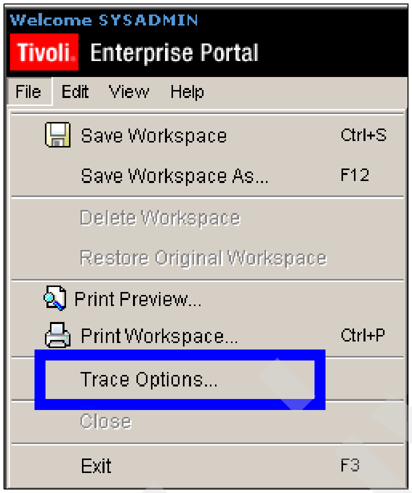


Figure 6-23 Trace Options

The Current trace selection field shows the current level of tracing.

2. Select a trace class from the list or as instructed by IBM Software Support (such as UNIT:TableAdapter ALL), as shown in Figure 6-24:
   - ALL provides data for all classes. Use this setting only temporarily, because it generates large amounts of data.
   - ERROR logs internal error conditions. This setting provides the minimum level of tracing, with little CPU processor usage, and ensures that program failures are caught and detailed.
   - NONE turns off the error log so that no data is collected.

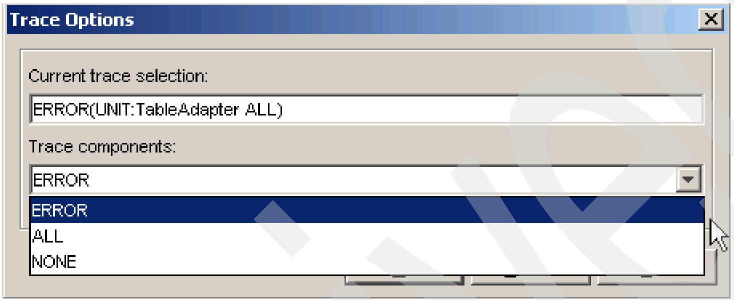


Figure 6-24 Tivoli Enterprise Portal Trace Options

3. Click **OK** to close the window and turn on logging.

6.10 IBM Tivoli Universal Agent troubleshooting

In this section, we describe some troubleshooting techniques for the IBM Tivoli Universal Agent. When troubleshooting the Tivoli Universal Agent, you have additional methods to trace problems.

6.10.1 Setting the trace

The IBM Tivoli Universal Agent uses the RAS1 trace that is written in the logs subdirectory. By default, the RAS1 trace has trace level of ERROR. We set the KDC_DEBUG variable to Y for yes in the KUMENV file in Windows and um.ini in UNIX systems. In Windows systems, the log file is located in *ITM_InstallDir*\TMAITM6\logs\KUMRAS1.LOG.

The KDC_DEBUG variable diagnoses communication problems between the Universal Agent and the Tivoli Enterprise Portal Server.

Note: Detailed RAS1 tracing might degrade the Universal Agent performance due to high CPU usage and input/output (I/O) activity.

Setting the IBM Tivoli Universal Agent trace involves the following steps:

1. In the Manage Tivoli Enterprise Monitoring Services window, right-click **Universal Agent** and select **Advanced** → **Edit Trace Parms**.
2. In the Universal Agent: Trace Parameters window, choose the appropriate filter for the trace log (Figure 6-25).

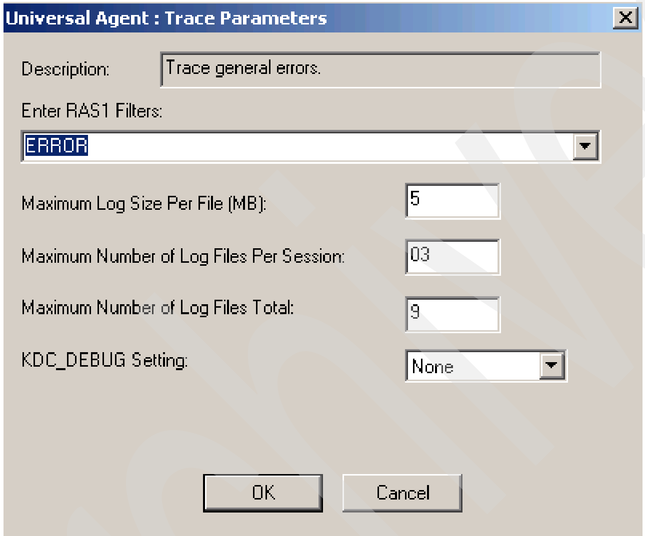


Figure 6-25 Setting the Trace Parameters

3. Restart the Universal Agent to implement the changes.

6.10.2 UAGENT application

The UAGENT application is a diagnostic tool that comes online during the data provider startup and comes with the DPLOG and ACTION workspaces. This application helps determine problems with the Universal Agent.

DPLOG

DPLOG is an event table that maintains the most recent 100 rows, unless you 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 is validated successfully, or if it failed validation (which means that its application will not come online).
- Whether a data source is available at startup.
- Which console ports and socket listening ports are used or are 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 was connected and disconnected.

ACTION

The ACTION table rows have a time-to-live value of 30 minutes. ACTION, different from the 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 6-26 shows the DPLOG workspace for the HTTP UAGENT application.

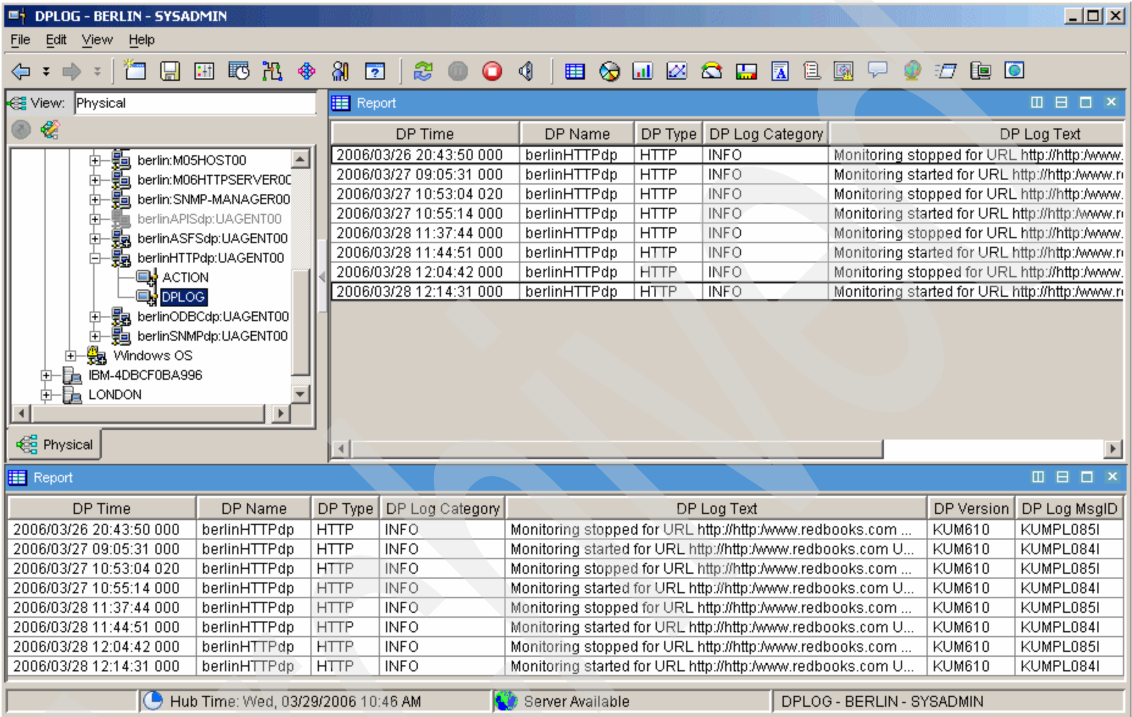


Figure 6-26 DPLOG workspace

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.

6.10.3 IBM Tivoli Universal Agent problem determination

Some of the 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, and then proceed to the other Tivoli Monitoring Services components.

Tips to determine problems relating to the 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 the Tivoli Enterprise Portal. It might include messages that help in 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 6-15 shows the error log indicating that the Universal Agent could not be started.

Example 6-15 RAS1 log

```
kum0sock.c,110,"KUM0_OpenLocalSocket") bind failed for local address UDP socket
512, port=1919, = error=10048 kum0sock.c,110, "KUM0_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, you 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. Then define a new port for the Universal Agent startup.

6.11 Working with IBM Support

For support for issues with IBM Tivoli Monitoring Express V6.1, the following information is helpful. Prepare these requests prior to placing your call so that you can be ready to submit the information to IBM Tivoli Software L2 support.

- The Microsoft Windows version and the service pack on the Tivoli Enterprise Portal client and Tivoli Enterprise Portal Server.
- Tivoli Enterprise Monitoring Server version and platform (determine this in Manage Tivoli Enterprise Monitoring Services).
- Running Tivoli Enterprise Portal in desktop or browser mode.
- If browser mode: Internet Explorer version at client.
- Agent types, versions, and where deployed.
- All necessary environmental information including the version of the Tivoli Enterprise Portal and the build level (from the Tivoli Enterprise Portal client kcjras1.log file).
- All logs from all the components involved.
- Tivoli Enterprise Monitoring Server log kmsras1, always necessary, might include additional information.

For a more comprehensive help support, perform the following steps:

1. Open Tivoli Enterprise Portal.

2. Click **Help** → **Work with IBM Support** (Figure 6-27). This enables you to refer to the following links:

   – IBM Tivoli Monitoring Software Support Web site
   – IBM Information Center Web site
   – Contacting IBM Software Support
   – Troubleshooting
   – Searching Knowledge Databases
   – Obtaining Fixes

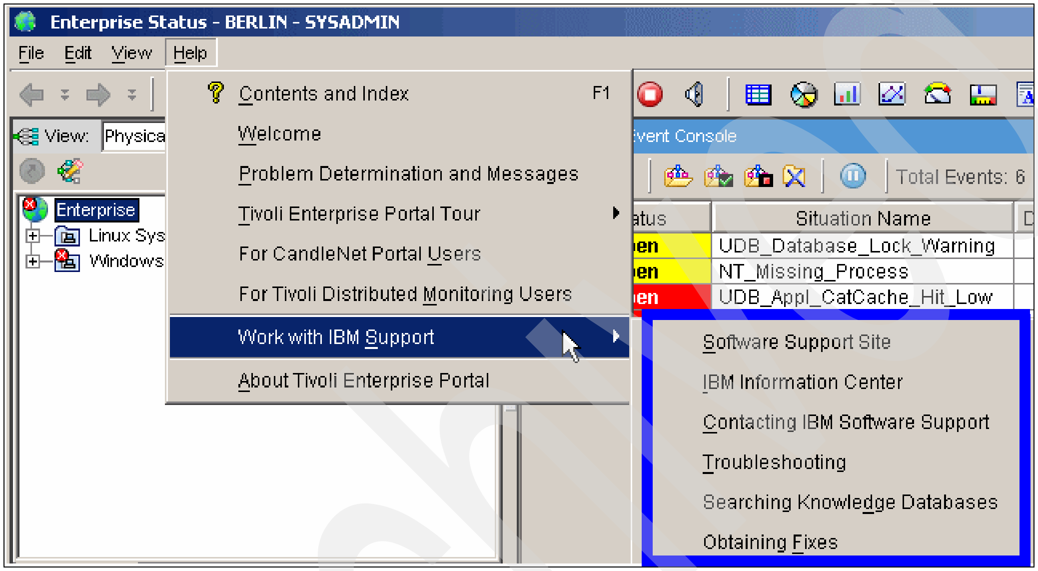


Figure 6-27 Work with IBM Support

A

Additional material

This IBM Redbook refers to an additional material that you can download from the Internet as described in this appendix.

Locating the Web material

The Web material associated with this redbook is available in softcopy on the Internet from the IBM Redbooks Web server. Point your Web browser to:

ftp://www.redbooks.ibm.com/redbooks/SG247217

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, SG247217.

Using the Web material

The additional Web material that accompanies this redbook includes the following file:

File name	*Description*
SG247217.zip	Zipped custom situations

System requirements for downloading the Web material

The following system configuration is recommended:

Hard disk space:	10 MB minimum
Operating system:	Microsoft 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 396. Note that some of the documents referenced here might be available in softcopy only.

- *Deployment Guide Series: IBM Tivoli Monitoring 6.1*, SG24-7188
- *Getting Started with IBM Tivoli Monitoring 6.1 on Distributed Environments*, SG24-7143

Other publications

These publications are also relevant as further information sources:

- *Getting Started with IBM Tivoli Monitoring Express*, SC32-1903
- *IBM Tivoli Monitoring for Active Directory User's Guide*, SC32-9444
- *IBM Tivoli Monitoring Universal Agent User's Guide*, SC32-9459
- *IBM Tivoli Monitoring V6.1 Administrator's Guide*, SC32-9408
- *IBM Tivoli Monitoring V6.1 i/5 OS Agent User's Guide*, SC32-9448
- *IBM Tivoli Monitoring V6.1 Installation and Setup Guide*, GC32-9407
- *IBM Tivoli Monitoring V6.1 Problem Determination Guide*, GC32-9458
- *IBM Tivoli Monitoring V6.1 User's Guide*, SC32-9409

Online resources

These Web sites and URLs are also relevant as further information sources:

- IBM Passport Advantage Web site

  http://www.ibm.com/software/sw-lotus/services/passport.nsf/%20WebDocs/Passport_Advantage_Home

- The IBM Solutions Consultant Express Tool Web site

  `http://www.ibm.com/partnerworld/solutionsbuilder`
- The Virtual Innovation Center Web site

  `http://www.ibm.com/partnerworld/vic`
- IBM Tivoli Open Process Automation Library for Business Partners Web site

  `http://www-18.lotus.com/wps/portal/tm`
- Gulf Breeze Software

  `http://gulfsoft.com`
- Gulf Breeze Software Web site for downloading sample custom situations

  `http://sourceforge.net/projects/gulfsoft`
- OpenESM project

  `http://sourceforge.net/projects/gulfsoft`
- Information about SOAP

  `http://w3schools.com`
  `http://w3.org/TR/SOAP`

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

Symbols

Numerics

A

B

C

U

V

W

X

Z

IBM

Redbooks

Deployment Guide Series: IBM Tivoli Monitoring Express Version 6.1

(0.5” spine)
0.475”<->0.875”
250 <-> 459 pages

Deployment Guide Series

IBM Tivoli Monitoring Express Version 6.1

Provides a step-by-step deployment guide for IBM Tivoli Monitoring Express

Discusses best practices for a deployment plan

Describes architecture and planning considerations

IBM Tivoli Monitoring Express Version 6.1 is a powerful, affordable, and easy-to-use availability management solution designed to help small to mid-sized companies manage IT infrastructures. It offers the ability to manage bottlenecks, performance impacts, and outages across heterogeneous environments from a single, centralized portal.

IBM Tivoli Monitoring Express V6.1 is easy to install, easy to deploy, and easy to use, providing rapid time to value. It provides real-time and historical data that enables you to quickly diagnose and solve issues with the new GUI through the IBM Tivoli Enterprise Portal component.

This IBM Redbook presents a deployment guide for IBM Tivoli Monitoring Express V6.1. We describe planning, installing, and troubleshooting IBM Tivoli Monitoring Express V6.1. In addition, we provide some case studies that you can use as part of a proof of concept or a customer demonstration.

The target audience for this book is IT specialists working on new IBM Tivoli Monitoring Express V6.1 installations.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-7217-00 ISBN 0738497444