Tivoli. software

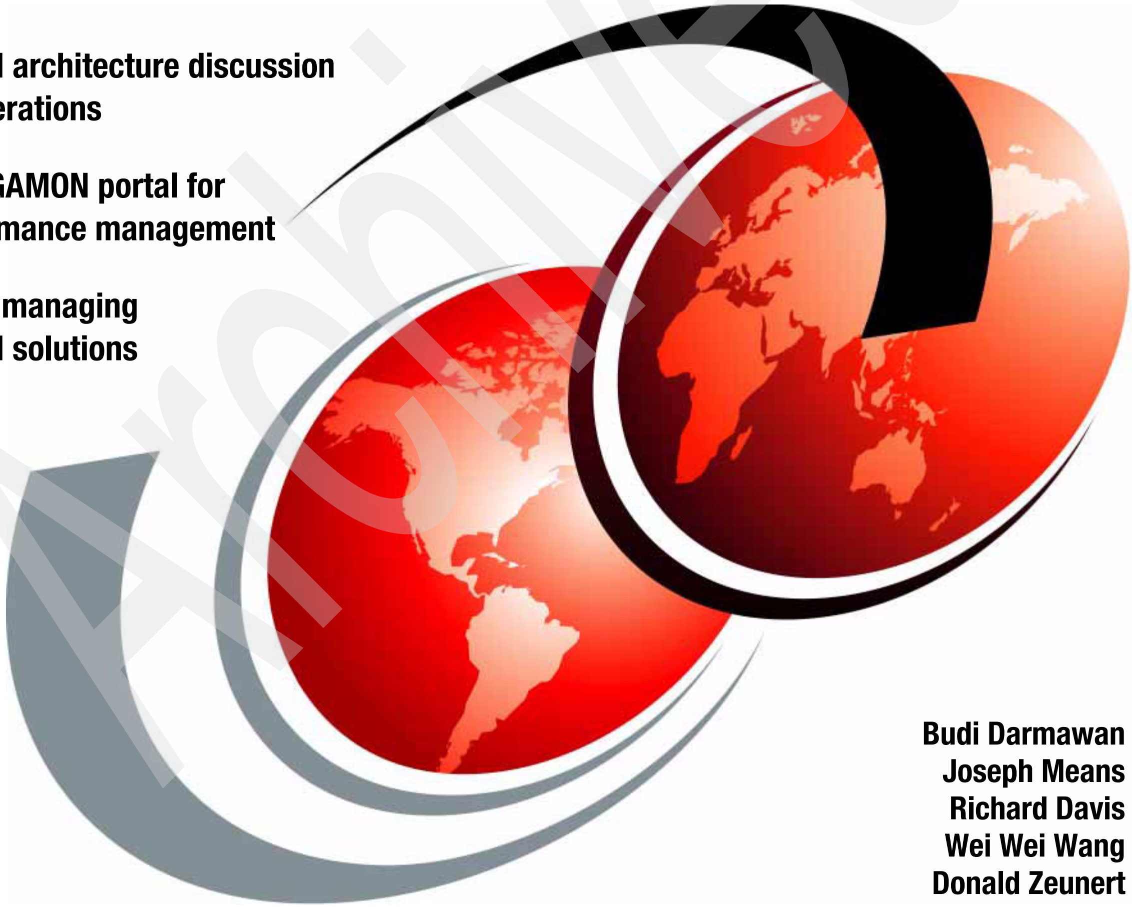
IBM

IBM Tivoli OMEGAMON XE V3.1 Deep Dive on z/OS

OMEGAMON architecture discussion and considerations

Using OMEGAMON portal for z/OS performance management

Tuning and managing OMEGAMON solutions



Budi Darmawan
Joseph Means
Richard Davis
Wei Wei Wang
Donald Zeunert

Redbooks

ibm.com/redbooks

International Technical Support Organization

IBM Tivoli OMEGAMON XE V3.1 – Deep Dive on z/OS

April 2006

SG24-7155-00

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

First Edition (April 2006)

This edition applies to Version 3.1.0 of the IBM Tivoli OMEGAMON XE family of products, including IBM Tivoli OMEGAMON XE on z/OS, IBM Tivoli OMEGAMON XE for CICS on z/OS, IBM Tivoli OMEGAMON XE for IMS on z/OS, IBM Tivoli OMEGAMON XE for Mainframe Network, and IBM Tivoli OMEGAMON XE for Storage on z/OS.

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®
- Candle®
- Candle Command Center®
- Candle Management Server®
- Candle Management Workstation®
- CandleNet®
- CandleNet eBusiness Platform®
- CandleNet Command Center®
- CandleNet Portal®
- Common User Access®
- CICS®
- CICSPlex®
- CL/CONFERENCE®
- CL/SUPERSESSION®
- CUA®
- DB2 Connect™
- DB2 Universal Database™
- DB2®
- DEXAN®
- eBA®
- eBusiness Assurance®
- EPILOG®
- ETE™
- ETEWatch®
- IntelliWatch®
- IBM®
- IMS™
- Lotus®
- MQSeries®
- MVS™
- NetView®
- OMEGACENTER®
- OMEGAMON II®
- OMEGAMON®
- OMEGAVIEW II®
- OMEGAVIEW®
- OS/2®
- OS/390®
- OS/400®
- Parallel Sysplex®
- Redbooks™
- Redbooks (logo) ™
- fRERETAIN®
- RMF™
- Tivoli Enterprise™
- Tivoli®
- TME 10™
- TME®
- VTAM®
- WebSphere®
- z/OS®
- zSeries®

The following terms are trademarks of other companies:

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

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

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

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

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

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

Preface

This IBM® Redbook describes the IBM Tivoli® OMEGAMON® XE product family, Version 3.1.0, that runs on z/OS®. We start by providing the product history over time and the terminology convention that we will use in the book, which should be noted carefully because the products are in transition from Candle® terminology to the new terminology related to IBM Tivoli Monitoring Version 6.1. We conform to IBM Tivoli Monitoring V6.1 terminology as much as possible, although for some direct references to screen shots, we use the terminology that is shown in the figure.

The product is installed using the IBM Configuration Tool. This is explained for new users who want to install IBM Tivoli OMEGAMON XE, because the process is somewhat different from standard IBM products that are installed on z/OS. The concept of runtime environment is explained here as well.

On the operation side of IBM Tivoli OMEGAMON XE, we discuss problem determination and tracing concepts and facilities that are available for IBM Tivoli OMEGAMON XE, as well as performance considerations. As IBM Tivoli OMEGAMON XE is meant to monitor performance of the system, its overhead should not affect overall system performance.

Last but not least, we offer sample usage scenarios of performance management using IBM Tivoli OMEGAMON XE products.

The team that wrote this redbook

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

Budi Darmawan is a Consulting IT Specialist at the International Technical Support Organization, Austin Center. He writes extensively and teaches IBM classes worldwide about all areas of performance and operational management with Tivoli solutions. Before joining the ITSO six years ago, Budi worked as solution architect and lead performer in IBM Global Services Indonesia. His current interests are in performance management, availability management, operational automation, and Java™ programming on both mainframe and distributed environments.

Joseph Means is a Consulting IT Specialist in the United States of America. He has more than 25 years of experience in Information Technology as an

Application Developer, Technical Support for Transaction/Database systems, Systems Engineer, Solutions Manager, and Account Solutions Engineer. Since the IBM acquisition of Candle Corporation in June 2004, he has worked with accounts to provide IBM Tivoli Performance Automation solutions that address client needs.

Richard Davis is an IT Specialist working in the UK Software Support Centre. He has more than 20 years of experience in the zSeries® arena. He has worked for IBM for three years, previously working as an IT consultant in the UK. His areas of expertise includes zSeries, WebSphere® on z/OS, and Lotus® Domino.

Wei Wei Wang is an Advisory IT Specialist working in the mainframe software support and services department of IBM Global Services China. She joined IBM in 2002 after she graduated from Southeast University, China, with a Master's degree in computer science. She gained well-rounded first-hand system management experience through leading on various Tivoli mainframe customer projects. Her areas of expertise includes z/OS, Parallel Sysplex®, storage, IBM Tivoli OMEGAMON XE, IBM Tivoli System Automation, IBM Tivoli Business System Management, and network and system security. She enjoys planning and developing system management solutions for mainframe and distributed platforms.

Donald Zeunert has worked in the Chicago (Oak Brook) regional office for the past 16 years. In his current position as a Senior Software Systems Engineer in the Tivoli Performance labs, he provides expertise to help ensure optimum use and effectiveness of OMEGAMON products on zSeries. Previously he was a pre-sales Consulting Systems Engineer at Candle for 13 years. He specializes in tuning of OMEGAMON products and subsystems for MVS™, USS, CICS®, DB2®, and MQSeries®. He was also a frequent speaker at SHARE and local CMG and NASPA meetings. Before joining Candle he had more than 12 years of experience in systems programming, performance management, database design, and application programming.

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

Robert Haimowitz, Richard M Conway, Roy P Costa, Betsy Thaggard
International Technical Support Organization

Armando Miramontes, Yoichiro Ishii, Robert Risley, Yvonne Scott-Matute, Nicola “Nic” Catrambone, Anthony Anderson, Richard Roy
IBM Software Group

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. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

1

Introducing IBM Tivoli OMEGAMON XE

This chapter introduces background information about performance management using the IBM Tivoli OMEGAMON XE. The discussion in this chapter consists of:

1.1 Performance management

The IT infrastructure of your enterprise supports many important business operations. Therefore, ensuring IT infrastructure reliability and availability is critical for business success. Given the increased complexity of IT infrastructures in general and e-business infrastructures in particular, intelligent, robust performance and availability management tools are essential for the proactive identification and resolution of problems before they affect business operations. You need tools that enable you to effectively monitor individual resource performance as well as overall business processes, while gauging resource functions and availability across your heterogenous IT infrastructure.

Back-end systems with a mainframe running z/OS have been the life-blood of many organizations, and although these mainframes have been designed to be robust and reliable, they still require management to ensure optimum performance, high availability, and superb reliability. Key performance management concerns regarding z/OS include:

- Ability to effectively manage a large number of systems and partitions with a limited number of operators.
- Early warning system to detect potential performance problems and perform corrective efforts, even before an outage happens.
- Common and uniform mechanism for performance management to limit switching systems and easier knowledge transfer for operators.
- Assistance in getting all of the lean power of the mainframe performance to ensure that all the overhead that occurs is essential.

1.2 IBM Tivoli OMEGAMON XE

IBM Tivoli OMEGAMON XE is an enterprise-class, easy-to-use set of solutions that optimize the performance and availability of your entire IT infrastructure. Monitoring is performed by the IBM Tivoli OMEGAMON XE family of products. IBM Tivoli OMEGAMON XE product family is a systems management integration tool for IT staff and management, with a capability to define the logical view of the systems in their enterprise and create processing logic to process situation events. The IBM Tivoli OMEGAMON XE product family enables you to see information from a wide variety of tools in one location, from multiple IBM Tivoli OMEGAMON XE monitors, as well as third-party software. IBM Tivoli OMEGAMON XE is designed to help you make decisions more quickly, efficiently, and proactively on an enterprise level.

Through a single customizable workspace portal, you can proactively manage the health and availability of your IT infrastructure, end-to-end, including operating systems, databases, and servers, across distributed and host environments. IBM Tivoli OMEGAMON XE detects bottlenecks and potential problems in essential system resources and helps you automatically recover from critical situations to ensure that your business-critical applications are up and running.

Built on a lightweight, highly scalable architecture, IBM Tivoli OMEGAMON XE is quickly deployed and easily managed for quick time to value and lower cost of ownership. Simple and centralized control, enhanced visualization of information, ease of use, and historical and real-time reporting enable users to quickly access the information they need, in customizable formats, to rapidly identify, diagnose, and resolve situations.

By linking IT services to processes, data, skills, and tools through its single user workspace, IBM Tivoli OMEGAMON XE enables users to view consistent data across technology domains and align IT services with business goals to deliver immediate value. IBM Tivoli OMEGAMON XE, when combined with IBM Tivoli Composite Application Manager and with IBM Tivoli Monitoring, offers complete, integrated availability management to ensure that your IT resources and staff are operating efficiently and effectively, in alignment with your business needs and priorities.

Table 1-1 highlights some common features of IBM Tivoli OMEGAMON XE.

Table 1-1 Facilities of IBM Tivoli OMEGAMON XE

Features	Advantages	Benefits
View information from multiple applications and monitors to understand the impact of your performance and availability issues	Quickly determine whether system performance problems are affecting high-priority applications	Designed to help you meet and exceed service levels by proactively managing critical applications
Combine information from multiple OMEGAMON monitors in a single workspace for an integrated view	Correlate transactions across mainframe and distributed and middleware environments	Enables you to quickly identify the source of application problem in a heterogeneous environment
Web browser interface enables you to monitor the health of your system	Common and familiar interface maintained across IBM Tivoli OMEGAMON XE interfaces	Designed to eliminate need for special training; navigation designed to avoid logging into monitors or products separately

Features	Advantages	Benefits
Graphic maps and icons can be easily integrated into management workspaces, enabling a quick and easy view of enterprise resources and applications, along with a status indication of the availability of those components	Can be used to create a global view of your data centers or a detailed system topology map of your local environment	Background maps are useful to provide a quick visual indication of where problems exist in a geographically dispersed enterprise monitoring environment
Graphical editor enables you to create alerts and complex thresholds, designed to be easily created without coding or deep mainframe knowledge	Includes thousands of data elements to define conditions attributable to system degradation	When alerts fire, you can quickly see what triggered the alert and get Expert Advice that explains the problem and offers potential fixes
Situation editor enables you to create complex thresholds, situations, and alerts	Most granular control available on finding performance problem	Use automation to respond promptly to problems without the need for deep mainframe scripting or coding skills
Expert Advice explains alerts and offers potential fixes; can be edited to include knowledge and solutions specific to your environment	Helps resolve recurring problems	Designed to help resolve problems and preserve knowledge for the entire staff, especially when tied into automation solutions such as reflex automation, automated alert escalation, and policy-based automation
Policy management / workflow editor is a graphic editor enabling you to create logic-based response and actions for raised situation events	Go beyond simple reflex automation capabilities by defining a series of actions, responses, additional actions to handle complex situations	Policy management / workflow editor enables you to quickly create automated responses to complex situation
Built-in TN3270 component	Drill down to see more granular details; navigate from the Web-based enterprise view to the details workspaces down to the TN3270 interface, all from a single point of control	Designed to allow for quick and painless migration from the OMEGAMON II® and classic interfaces to OMEGAMON
Built-in Web browser component	Enables you to include information from other Web browser interfaces within workspaces	Designed to allow quick integration of key metrics from browser-interfaced sources

Features	Advantages	Benefits
Built-in ODBC query component	Enables you to include information from ODBC sources	Designed to allow quick integration of key indicators and information (IT-related or other business metrics or information) into IBM Tivoli OMEGAMON XE workspaces

This section looks at features of the IBM Tivoli OMEGAMON XE product family:

- IBM Tivoli OMEGAMON XE on z/OS V3.1.0
- IBM Tivoli OMEGAMON XE for CICS on z/OS V3.1.0
- IBM Tivoli OMEGAMON XE for DB2 on z/OS V3.1.0
- IBM Tivoli OMEGAMON XE for IMS on z/OS V3.1.0
- IBM Tivoli OMEGAMON XE for Mainframe Networks V3.1.0
- IBM Tivoli OMEGAMON XE for Storage on z/OS V3.1.0

1.2.1 IBM Tivoli OMEGAMON XE on z/OS V3.1.0

IBM Tivoli OMEGAMON XE on z/OS V3.1.0 combines the monitoring capabilities of OMEGAMON for Sysplex, OMEGAMON for OS/390®, and OMEGAMON for IBM Cryptographic Coprocessors. This product enables you to effectively monitor the availability, performance, and resource utilization of sysplexes and the individual z/OS systems that participate in them.

IBM Tivoli OMEGAMON XE on z/OS provides comprehensive information about sysplex-level components such as coupling facilities, global enqueue, global resource serialization (GRS) ring systems, report classes, service classes, resource groups, shared direct access storage device (DASD) groups, and cross-system coupling facilities. It also provides extensive system-level information about the z/OS images in the sysplexes and monitors the status and configuration of IBM cryptographic coprocessors installed in zSeries servers.

IBM Tivoli OMEGAMON XE on z/OS also provides a powerful analytic interface to classic OMEGAMON and OMEGAMON II monitoring products. Used in conjunction with other OMEGAMON monitoring products, the data, analyses, and alerts presented by IBM Tivoli OMEGAMON XE on z/OS help you develop a holistic view of your entire computing enterprise from a single console. Along with the important information OMEGAMON II provides, IBM Tivoli OMEGAMON XE on z/OS adds the features and functionality offered by the IBM Tivoli Monitoring Services.

IBM Tivoli OMEGAMON XE on z/OS has the following features:

- Supports zSeries hardware.
- Provides information about resource usage.

- Monitors TSO users and batch workloads, CPU utilization, memory, paging, disk.
- I/O performance, and LPAR logical and physical configuration.
- Performs subsystem workflow analysis to determine which resources a service class is using and which resources, including lock waits, have delayed service class performance.
- Analyzes batch performance and disk control units and caches.
- Identifies excessive wait times through bottleneck analysis.
- Analyzes the delays caused by competing resources through impact analysis.
- Performs response-time analysis.
- Provides real-time performance metrics for IBM Cryptographic Coprocessors.
- Has an interface that enables you to issue operator commands and take corrective actions.

1.2.2 IBM Tivoli OMEGAMON XE for CICS on z/OS V3.1.0

IBM Tivoli OMEGAMON XE for CICS on z/OS helps you proactively manage complex CICS systems (including CICS in an IBM Parallel Sysplex environment) to achieve high performance and avoid costly downtime. With a flexible, easy-to-use browser interface, IBM Tivoli OMEGAMON XE for CICS on z/OS helps you clearly see and understand application and system events. You can monitor and manage CICS transactions at the big-picture and granular levels, as well as interaction with other applications, within a single interface. IBM Tivoli OMEGAMON XE for CICS on z/OS is designed to enable you to detect problems quickly and take action in real time to speed problem resolution.

IBM Tivoli OMEGAMON XE for CICS on z/OS does the following:

- Proactively manages performance and availability of IBM CICS systems from a single, integrated interface.
- Monitors transactions to understand how resource usage and performance affect internal users and customers. Identifies tasks waiting for specific resources and pinpoints excessive wait times to resolve issues quickly.
- Monitors virtual storage access method (VSAM) files, identifies record-level sharing locks, and facilitates rapid resolution to maximize availability.
- Correlates CICS log streams with associated facility structures to fine-tune CICS systems for optimal performance.
- Exploits CICS Transaction Server monitoring.

1.2.3 IBM Tivoli OMEGAMON XE for DB2 on z/OS V3.1.0

IBM Tivoli OMEGAMON XE for DB2 Performance Monitor on z/OS and IBM Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS helps you proactively manage your DB2 mainframe environment and tune for optimal performance. These products bring the strength of both OMEGAMON for DB2 and DB2 Performance Monitor/Expert into a new set of DB2 monitors. The difference between the Monitor and the Expert product offering is that the Expert provides additional expert analysis functions such as buffer pool analysis and simulation, as well as specific expert rule-of-thumb and SQL performance queries executed against data stored in the performance warehouse.

The products achieve this robustness by tapping into DB2 Version 8 features, such as 64-bit above-the-bar virtual addressing and long SQL statements, and advancing distributed thread tracking, workload monitoring, historical reporting, and more. The Web interface provides a single interface at the big-picture and granular levels, including interaction between DB2 and other applications. This solution helps you identify performance spikes and anomalies that might otherwise go unseen, take action in real time, and automate repetitive DB2 operations.

IBM Tivoli OMEGAMON XE for DB2 on z/OS has the following features:

- Monitors threads, I/O, buffer pools, SQL cache.
- Views threads running stored procedures.
- Monitors interaction of DB2 with subsystems such as CICS and IMS™.
- Includes an application trace facility that tracks every step of a DB2 transaction.
- Monitors DB2 Connect™ and the DCS applications.
- Provides in-depth analyis with detailed reports and the Performance Warehouse with its analysis support.
- Provides buffer pool analysis and simulation with the Performance Expert product.

1.2.4 IBM Tivoli OMEGAMON XE for IMS on z/OS V3.1.0

IBM Tivoli OMEGAMON XE for IMS on z/OS helps you optimize the performance and availability of your vital IMS systems. From a single point of control, you can view comprehensive information and analysis across multiple IMS subsystems, or across your IMSplex environment. With IBM Tivoli OMEGAMON XE for IMS on z/OS, you can collect and summarize information about key resources, such as enqueue, I/O, CPU, paging rates, pool storage, and buffer pool metrics. The product provides both granular and system-wide views of your IMS operations, giving you comprehensive information and analysis across multiple IMS subsystems or across your IMSplex environment.

IBM Tivoli OMEGAMON XE for IMS on z/OS has the following features:

- Views IMS Pools showing utilization, pool storage sizes, and amount of free blocks.
- Auto-discovers IMS and Internal Resource Lock Manager (IRLM) subsystems.
- Enables you to view coupling facility statistics to identify factors affecting performance.
- Monitors workload balancing using shared queue support and data sharing to minimize the impact of locks on shared databases.
- Tracks and optimizes both resource usage and transaction processing.
- Monitors resource usage for IMS regions with detailed metrics such as CPU usage, I/O activity, storage, paging, and Execute Channel Program (EXCP).

1.2.5 IBM Tivoli OMEGAMON XE for Mainframe Networks V3.1.0

IBM Tivoli OMEGAMON XE for Mainframe Networks gives you the ability to monitor and manage the health of crucial TCP/IP connections (in addition to the traditional VTAM® and NCP operations provided by the OMEGAMON II for Mainframe Networks component) within your IBM z/OS environments. IBM Tivoli OMEGAMON XE for Mainframe Networks is designed to analyze TCP/IP performance among CICS, DB2, IMS, and other key applications to identify resource loss and unstable connections. The power to monitor the end-user experience lets you fine-tune TCP/IP performance before customers complain.

IBM Tivoli OMEGAMON XE for Mainframe Networks has the following features:

- Monitors VTAM sessions and TCP/IP.
- Monitors file transfer protocol (FTP) and TN3270 sessions.
- Monitors High-Performance Router (HPR) links and Enterprise Extender (EE) connections.
- Monitors performance of OSA adapters.
- Provides information about application connections, ports, packet size, and volume.
- Measures IP discards, reassemblies, and fragmentation.
- Provides detailed analysis of TCP/IP connections that are slow, unstable, or down.

- Displays information for each TCP/IP connection such as application name and IP addresses.
- Measures transaction round-trip time using the OMEGAMON II for Mainframe Networks ETE™ component.

1.2.6 IBM Tivoli OMEGAMON XE for Storage on z/OS V3.1.0

IBM Tivoli OMEGAMON XE for Storage on z/OS combines comprehensive storage performance monitoring with a flexible, easy-to-use browser interface that helps you to be more productive, to more clearly understand storage conditions, and to ensure optimal performance. The product is designed to manage the performance and availability of mainframe-attached storage (including DASD and tape devices) and the datasets that reside on them. It also features in-depth analysis of two important IBM storage software components: Data Facility Systems Managed Storage (DFSMS), which manages the service levels and priorities of data sets based on user-created storage goals; and Data Facility Hierarchical Managed Storage (DFHSM), which manages backup of data based on usage patterns. IBM Tivoli OMEGAMON XE for Storage on z/OS features an easy-to-use browser-based interface, online trending and historical information, and the ability to monitor user-created groups of DASD volumes.

IBM Tivoli OMEGAMON XE for Storage on z/OS has the following features:

- Manages performance and availability of mainframe-attached disk and tape storage, and datasets on that media.
- Displays capacity alerts to show space available within volume and storage groups.
- Provides in-depth analysis of channel, tape, DF/SMS, and DF/HSM.
- Provides volume statistics and performance metrics such as DASD fast write (DFW) ratios, cache hit rates, volume contention, resource locks, and data collisions.
- Monitors channel utilization, dataset-level performance, and space metrics.
- Identifies and monitors the datasets that an application is currently using.

1.3 History of IBM Tivoli OMEGAMON XE monitoring

This section provides a brief overview of the OMEGAMON history. The discussion is divided into the following time-line:

- 1.3.1, “The mid-1970s through the mid-1980s” on page 10
- 1.3.2, “The late 1980s through the early 1990s” on page 10

1.3.1 The mid-1970s through the mid-1980s

In 1976 Aubrey Chernick released OMEGAMON for MVS (“the last monitor you will ever need”). For the first time, this introduced software monitoring paradigm versus hardware monitoring. The cost of the solution could be justified by delaying expensive hardware upgrades for months. The user interface was a simple 3270 command interface that consisted of major and minor mnemonic commands that made its usage difficult to use for anyone but the very technical.

During this timeframe, many capabilities were developed that are still available today. This period represents the shift in importance from data collection to data analysis.

- Exception analysis is created, enabling OMEGAMON to warn when anomalies existed.
- EPILOG® (historical collection), DEXAN® (bottleneck analysis), and RTA (Response Time Analysis) released as separately licensed add-on components.
- The logical tuning approach is created by Aubrey Chernick: Identify - Isolate - Resolve.
- Commands are introduced to enable the user to resolve problems, such as:
  - The MZAP command to modify the contents of the common storage area
  - The KILL command to terminate an address space
- OMEGAMON for VM and OMEGAMON for VSE are introduced.
- OMEGAMON for CICS, OMEGAMON for IMS, and OMEGAMON for DB2 are introduced to provide application monitoring.
- The 3270 menu interface is released to simplify usage and enable broader audience appeal. The OMEGAMON monitors become Operations and Help Desk tools.

1.3.2 The late 1980s through the early 1990s

From the late 1980s through early 1990s, niche competitors started offering solutions with easy-to-use interfaces. To combat this competition and to improve usability, Candle purchased the CL technology to enable delivery of CUA® (Common User Access®) compliant 3270 interfaces.

- CL/Suppersession and CL/Conference are now sold by Candle.

- The OMEGAMON II products are released. Separate add-on components such as EPILOG, DEXAN, and RTA are included as part of OMEGAMON II. The term “keep the lights green” is born.
- OMEGAMON II for Mainframe Networks (originally named OMEGAMON II for VTAM) is released. This monitor was CUA only and built on CL technology, and did not contain an OMEGAMON classic (or collector) component.
- Many computer rooms are beginning to run out of screen real estate due to the number of OMEGAMON consoles needed. Candle-built Status Monitor uses CL as the first single pane of glass integration.
- In response to customer requests relative to Status Monitor overhead and panel customization capability, OMEGAVIEW® 3270 is launched.
- AF/Operator and AF/Remote are purchased to provide automation capabilities. APIs are created that integrate AF/Operator, OMEGAVIEW, and the OMEGAMONs. This integration provides the ability to define user-created status items and automate actions when performance problems occur.
- Candle acquires CDB to market DB2 tools: DB/Explain, DB/Workbench, DB/QuickChange, DB/SMU, and DB/DASD.

1.3.3 The mid-1990s

During the mid-1990s, multiple market forces were active in the system management area.

- AF/Operator is repackaged into OMEGACENTER® Gateway with OMEGACENTER Status Manager (OSM) to provide integration between OMEGAMON and other automation tools.
- Candle Command Center® (CCC) is released on OS/2® with agent/server infrastructure. Distributed operating systems and distributed database agents were offered for first time.
- OMEGAMON II for Storage (originally named OMEGAMON II for SMS) is released. This monitor was CUA only and built on CL technology and did not contain an OMEGAMON classic (or collector) component.
- Candle Command Center object-oriented interface is ported to Windows®, maintaining the OS/2 look and feel.
- Candle is integrated with Frameworks using alert adapters and alert emitters (AA/AEs).
- Candle releases Candle Command Center for MQSeries, which becomes the market share leader for MQSeries monitoring.

- Candle realizes that OS/2 object-oriented interface on Windows is not easy to use and begins development of the next generation of graphical user interface (GUI), which comes to be called OMEGAMON XE.
- Candle buys Integrated Resource Manager (IRM) from Bank of America and markets IRM as a service. IRM is designed to simplify automation implementation and integration with OMEGAVIEW.
- Candle releases Candle Command Center for UNIX® System Services.

1.3.4 The early 2000s

The early 2000s were turbulent times for Information Technology, as the impact of Year 2000 spending was felt across all of IT.

- Candle releases OMEGAMON for Cryptographic Coprocessor.
- eBusiness Assurance® (eBA®), eBusiness Assurance Network (eBAN), and ETEWatch® created to provide end user experience response time monitoring.
- OMEGAMON Extended Edition (XE) and OMEGAMON Dashboard Edition (DE) are released.
- More OMEGAMONs are released: OMEGAMON for WebSphere Application Server, OMEGAMON for WebSphere Interchange Server, and OMEGAMON for WebSphere Business Integration (WBI). All of these solutions used the OMEGAMON GUI only.
- Web Response Monitor (WRM) and Web Segment Analyzer (WSA) are released into the market.

1.3.5 IBM acquires Candle Corporation

On 01 April 2004, IBM announced the intent to acquire Candle Corporationl, and the acquisition went through on 08 June 2004. Subsequent to the acquisition, IBM Tivoli developed a roadmap to outline the directions for Tivoli and the former Candle solutions that contained overlapping or similar capabilities.

- OMEGAMON for CICS, OMEGAMON for CICSPlex®, and OMEGAMON II for CICS are combined into a single solution, IBM Tivoli OMEGAMON XE for CICS on z/OS.
- OMEGAMON for DB2, OMEGAMON for DB2Plex, and OMEGAMON II for DB2 are already combined into IBM Tivoli OMEGAMON XE for DB2 on z/OS. IBM Tivoli OMEGAMON XE for DB2 on z/OS, DB2 Performance Monitor, and DB2 Performance Expert are being merged into two solutions: IBM Tivoli OMEGAMON XE for DB2 PM and IBM Tivoli OMEGAMON XE for DB2 PE.

- OMEGAMON for IMS, OMEGAMON for IMSPlex, and OMEGAMON II for IMS are combined into a single solution: IBM Tivoli OMEGAMON XE for IMS on z/OS.
- OMEGAMON for Mainframe Networks, OMEGAMON II for Mainframe Networks, and IBM Tivoli Monitoring for Network Performance are combined into a single solution: IBM Tivoli OMEGAMON XE for Mainframe Networks. This entitlement will also include NetView® Performance Monitor (NPM).
- OMEGAMON for Storage becomes IBM Tivoli OMEGAMON XE for Storage.
- OMEGAMON for Sysplex, OMEGAMON for OS/390, OMEGAMON II for MVS, and OMEGAMON XE for Cryptographic Coprocessors are combined into a single solution: IBM Tivoli OMEGAMON XE for z/OS. OMEGAMON for UNIX System Services (USS) is currently a stand-alone solution. The current roadmap has OMEGAMON for USS included in the IBM Tivoli OMEGAMON XE for z/OS solution in the future.
- Other OMEGAMON solutions providing capabilities associated with WebSphere and distributed platforms are being enhanced and merged as well. (These solutions are beyond the scope of discussion in this book.)

1.4 Terminology convention

This book uses the new terminologies that were adopted with IBM Tivoli Monitoring V6.1 products as the result of the IBM acquisition of Candle. Although in some cases the screen shots and log files show the previous terms, we use the new terminology in the context of this book unless we are referring to a specific items in a screen shot. The following terms are used:

Tivoli Enterprise Monitoring Server
Formerly Candle Management Server® (or CMS or TEMS), this is the central piece of IBM Tivoli OMEGAMON XE architecture that acts as the data collection and control point.

Tivoli Enterprise Portal Server
Formerly Candle Net Portal Server (or CNPS or TEPS), this is the user interface server for the portal based interface that retrieve data from Tivoli Enterprise Monitoring Server.

Tivoli Enterprise Portal
Formerly Candle Net Portal (or CNP or TEP); the main user interface for IBM Tivoli OMEGAMON XE platform, the portal can be running as a native Java application or a browser-based applet.

Tivoli Enterprise Monitoring Agent
Intelligent Remote Agent (IRA) or OMEGAMON monitoring agent (OMA); performs that actual resource monitoring.

IBM Tivoli Monitoring Services
This term replaces the OMEGAMON platform, and typically refers to a collection of Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server and Tivoli Enterprise Portal.

ITMS:Engine
This term replaces the CT/Engine that represents the common architecture behind IBM Tivoli OMEGAMON XE started tasks and address spaces on z/OS.

IBM Configuration Tool
Refers to the ISPF panels and dialogs that make configuration of IBM Tivoli OMEGAMON XE easier. This replaces CICAT (Candle Installation and Configuration Assistant Tool).

Tivoli Data Warehouse
This provides persistent historical data storage for a period of more than 24 hours. The data warehouse is formerly called Candle Data Warehouse.

We will discuss detailed components, functions and architecture in Chapter 2, "IBM Tivoli OMEGAMON XE components and architecture" on page 19.

1.5 Product features

This section discusses an overview of the product components of the IBM Tivoli OMEGAMON XE solution:

- 1.5.1, "IBM Tivoli Monitoring Services" on page 14
- 1.5.2, "IBM Tivoli OMEGAMON XE" on page 15
- 1.5.3, "Interface to other Tivoli products on z/OS" on page 17

1.5.1 IBM Tivoli Monitoring Services

IBM Tivoli Monitoring Services is a set of components that serve as the interface for IBM Tivoli OMEGAMON XE products. This platform typically includes:

- Client side, which consists of:
  – Tivoli Enterprise Portal, which is a Java-based user interface for viewing and monitoring your enterprise network.

  - Candle Management Workstation® (CMW), which is the client component of a CandleNet® Command Center® (CCC) environment, which was a processor to OMEGAMON XE. It has been mostly replaced by the Tivoli Enterprise Portal user interface, but is required for performing some specific functions such as specifying CICSplex workload definitions.
- Tivoli Enterprise Portal Server, which manages connections and serves Tivoli Enterprise Portal users.
- Tivoli Enterprise Monitoring Server, which serves as the data collector and primary interface for IBM Tivoli OMEGAMON XE products.

Tivoli Enterprise Monitoring Agents are installed on the systems or subsystems whose applications and resources you want to monitor. The agent collects monitoring data from the managed system and passes it to the Tivoli Enterprise Monitoring Server to which it is connected. The client gathers the current values of the attributes and produces reports formatted into tables and charts. It can also test the values against a threshold and display an alert icon when that threshold is exceeded or a value is matched. These tests are called *situations*.

1.5.2 IBM Tivoli OMEGAMON XE

IBM Tivoli OMEGAMON XE is a suite of solutions that assist in monitoring mainframe systems. It provides a way to monitor the availability and performance of all systems in an enterprise from one or several designated workstations. It provides many useful workspaces that can be used to track trends and understand and troubleshoot system problems. The current IBM Tivoli OMEGAMON XE product is also associated with IBM Tivoli OMEGAMON DE for z/OS.

- IBM Tivoli OMEGAMON XE provides the monitoring capability in the form of Tivoli Enterprise Monitoring Agent that runs on the managed systems and subsystems. You can use OMEGAMON to:
  - Establish your own performance thresholds.
  - Create situations (conditions to monitor create and send commands to systems in your managed enterprise) by means of the Take Action feature.
  - Create comprehensive reports about system conditions.
  - Monitor for alerts on the systems and platforms you manage.
  - Trace the causes leading up to an alert.
  - Define your own queries, using the attributes provided by IBM Tivoli OMEGAMON XE to monitor conditions of particular interest to you.

- IBM Tivoli OMEGAMON DE for z/OS enhances IBM Tivoli OMEGAMON XE portal interface. The DE stands for dashboard edition. It adds these application integration components:
  - Multiple applications in one workspace

    In one workspace you can build a table or chart with data from one type of monitoring agent, and another table or chart with data from a different agent, showing views from as many different agent types as are included on that branch of the Navigator.
  - Linking between workspaces and Navigators

    You can define a link from a workspace associated with one application to the workspace of another, and you can link between Navigators.
  - Graphic view

    The Graphic view lets you place icons of Navigator items on a map or other background graphic, giving you a visual orientation for quick detection of problems anywhere in your monitored enterprise. You can zoom in, and you can add such graphics as floor plans and organization charts.
  - Custom Navigator views

    Tivoli Enterprise Portal comes with the navigator's Physical view, which organizes your managed enterprise by operating platform and agent type. The navigator logical view enables you to define your own hierarchy or grouping of managed objects. You can also define Navigator views for any logical hierarchy, such as a departmental or site hierarchy.
  - Event Console view

    In a single view, you can see the entire list of alerts that have occurred over a certain period. The event Console view lists each alert, including such information as the age of the alert, where and when it occurred, and the situation name. The console has a toolbar for filtering the view and a pop-up menu for managing alerts.
  - Automation

    The Workflow editor enables you to design sets of automated system processes, called policies, to resolve system problems. A policy performs actions, schedules work to be performed by users, or automates manual tasks.

1.5.3 Interface to other Tivoli products on z/OS

Tivoli products on z/OS that have interfaces to the current IBM Tivoli OMEGAMON XE are:

- IBM Tivoli NetView for z/OS Version 5.2 provides a Universal Agent component that enables NetView for z/OS measurement to be displayed in the Tivoli Enterprise Portal.
- IBM Tivoli Business Systems Manager V3.1 enables collection of IBM Tivoli OMEGAMON XE monitors as exception on z/OS based monitor. Latest fixpack of IBM Tivoli Business Systems Manager V3.1 also allows collection of IBM Tivoli OMEGAMON XE objects from the Tivoli Enterprise Monitoring Server level to be propagated directly to IBM Tivoli Business Systems Manager.

1.6 Project environment

The project took place at the International Technical Support Organization in Austin, with access to mainframe systems in a sysplex environment in Poughkeepsie. In our configuration, we rebuilt the system several times to demonstrate use of several key technologies. The primary managed environment is a set of three z/OS machines as shown in Figure 1-1, which also shows the IBM Tivoli OMEGAMON XE products that were installed in the system.

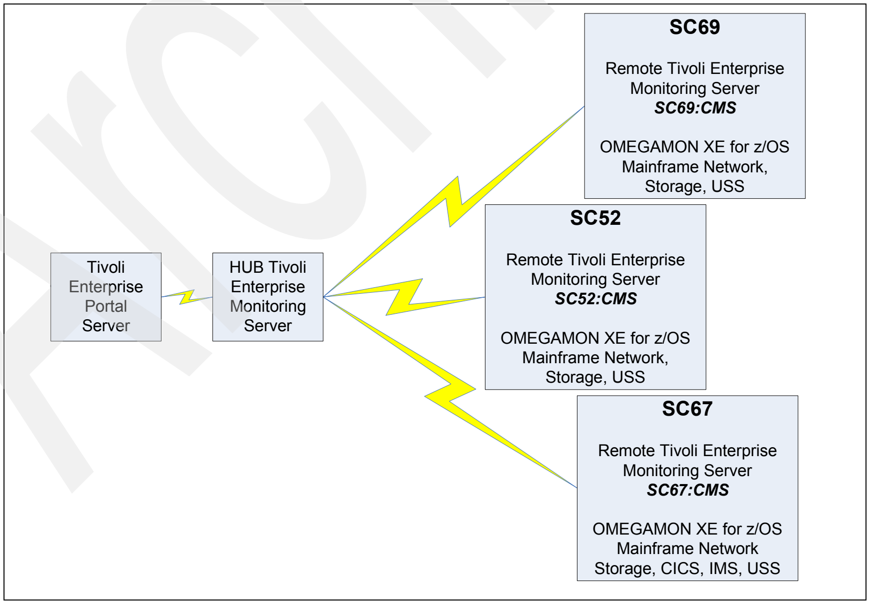


Figure 1-1 Project environment

We worked with several configurations to test the following scenarios:

- Running the hub on z/OS in SC52:CMS, the Tivoli Enterprise Portal Server was a Windows server.
- Building a movable hub on z/OS with all of the products Tivoli Enterprise Monitoring Server are running as remotes.
- Using IBM Tivoli Monitoring V6 Tivoli Enterprise Monitoring Server and Tivoli Enterprise Portal Server running on a Linux® on zSeries platform.

1.7 Document organization

The rest of this book consists of the following chapters:

- Chapter 2, “IBM Tivoli OMEGAMON XE components and architecture” on page 19 explains the concepts behind the IBM Tivoli OMEGAMON XE product structure.
- Chapter 3, “Installation and configuration: IBM Tivoli OMEGAMON XE” on page 39 provides some considerations and explanations about the installation process of IBM Tivoli OMEGAMON XE, including the navigation of the IBM Configuration Tool and runtime environment creation.
- Chapter 4, “Maintenance and problem determination” on page 113 shows some tracing techniques and data collection for problem determination.
- Chapter 5, “IBM Tivoli OMEGAMON XE performance optimization” on page 135 explains the detailed performance considerations for the IBM Tivoli OMEGAMON XE as a monitoring tool. IBM Tivoli OMEGAMON XE should not become a significant overhead for the overall system it monitors.
- Chapter 6, “Working with Tivoli Enterprise Portal” on page 191 describes some sample scenarios for working with IBM Tivoli OMEGAMON XE products.

2

IBM Tivoli OMEGAMON XE components and architecture

In this chapter, we introduce the components and architecture of IBM Tivoli OMEGAMON XE products. The discussion covers a wide range of topics from configuration options to failover. The chapter consists of:

2.1 Architecture and components

This section explains the overall architecture of IBM Tivoli OMEGAMON XE. The architecture is based on IBM Tivoli OMEGAMON XE V3.1. In this section we explain the structure and mechanism of IBM Tivoli OMEGAMON XE monitoring as well as the interrelationship and dependencies of its main components. The discussion in this section consists of:

- 2.1.1, “Architecture of IBM Tivoli OMEGAMON XE” on page 20
- 2.1.2, “IBM Tivoli Monitoring Services components” on page 24
- 2.1.3, “OMEGAMON components on z/OS” on page 26

2.1.1 Architecture of IBM Tivoli OMEGAMON XE

IBM Tivoli OMEGAMON XE products on z/OS consist of several independent components that make up the overall monitoring environment as shown in Figure 2-1.

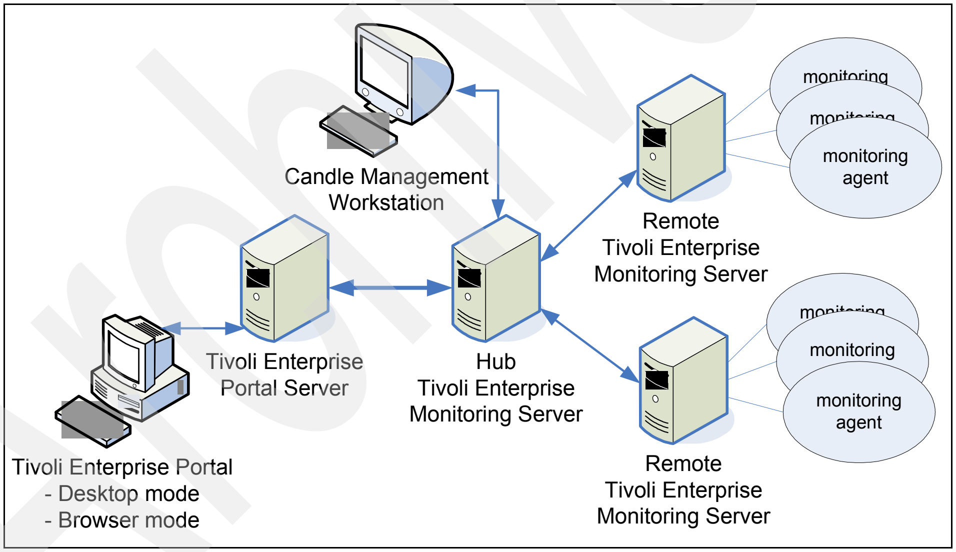


Figure 2-1 IBM Tivoli OMEGAMON XE platform architecture

As shown in Figure 2-1, the components of IBM Tivoli OMEGAMON XE are:

- Tivoli Enterprise Portal, a Java-based user interface that communicates with Tivoli Enterprise Portal Server to request and retrieve monitoring data. Depending on how it is installed, it enables you to view the data using a desktop application (desktop mode) or Web browser (browser mode).

- Tivoli Enterprise Portal Server, a Java application server that enables retrieval, manipulation, and analysis of data collected from the monitoring agents. It functions as a repository for all user data, such as user IDs, workspaces, views, and queries, which are used to retrieve data from different data sources such as agents. It connects to the hub Tivoli Enterprise™ Monitoring Server and can be accessed by the Tivoli Enterprise Portal client.
- Tivoli Enterprise Monitoring Server has these functions:
  - Consolidates the data collected by the agent and distributes the data either to the Tivoli Enterprise Portal Server or to the Hub Tivoli Enterprise Monitoring Server if it is a remote Tivoli Enterprise Monitoring Server.
  - Evaluates column function for situations that cannot be resolved at the Tivoli Enterprise Monitoring Agent.
  - Sends an alert to the Tivoli Enterprise Portal Server when specified conditions are met and receives commands from the interface and distribute them to the appropriate monitoring agents.
  - Stores historical data and prototypes for configuration in the form of *seed* data, which consists of product-specific situations, templates, and other sample data added to the Tivoli Enterprise Monitoring Server Enterprise Information Base (EIB) tables when a monitoring agent is installed.
  - Specify security settings such as logon password requirement.

  We discuss the mechanism of Tivoli Enterprise Monitoring Server in 2.1.2, "IBM Tivoli Monitoring Services components" on page 24.
- Candle Management Workstation (CMW) is the client component of a CandleNet Command Center environment. It has been mostly replaced by the Tivoli Enterprise Portal user interface, but is required for some advanced functions such as importing and exporting situations and policies, and support for Tivoli Enterprise Monitoring Server hot standby (hub switching).
- Tivoli Enterprise Monitoring Agent is responsible for monitoring and collecting performance data from your systems. These agents run on the managed systems, evaluating situations and returning only rows that match the criteria to Tivoli Enterprise Monitoring Server. Actions can be performed locally on Tivoli Enterprise Monitoring Agent. These agents also respond to and process queries from Tivoli Enterprise Portal Server. These agents can issue commands to the monitored system or application. This book discusses the monitoring agent specifically for z/OS environment.

Specific to z/OS implementation, which is the basis of IBM Tivoli OMEGAMON XE architecture, there are additional components that come from the history of the product, as shown in Figure 2-2 on page 22.

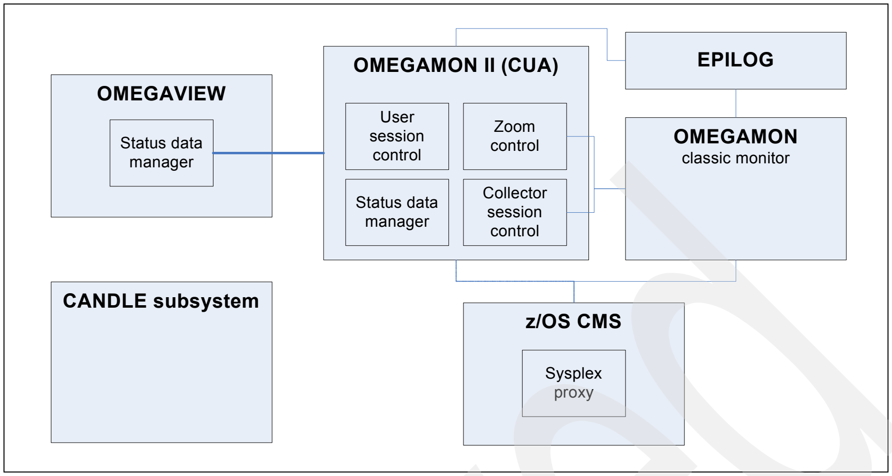


Figure 2-2 IBM Tivoli OMEGAMON XE architecture on z/OS

The components shown in Figure 2-2 are:

- OMEGAMON classic

  OMEGAMON classic collector collects information from the monitored platform. This is performed by setting user exits, evaluating in storage control blocks or calling system APIs. The OMEGAMON classic has a 3270 menu interface for managing the system. These OMEGAMON classic monitors are the basic monitoring systems for the IBM Tivoli OMEGAMON XE platform.

- OMEGAMON II

  OMEGAMON II is the 3270 status display that shows resources' status using a Common User Access (CUA) compliant color-coded application. This application collects status information and presents it from Status Data Manager. The status information comes from a variety of sources:

  – DB2,CICS, and IMS use the classic collector information.

  – Storage and Mainframe Network does not have a classic component, so the CUA collects all information.

  – MVS address space information is collected from both classic and CUA codes.

  – Workload Manager and Sysplex DASD information is collected from Tivoli Enterprise Monitoring Server; however, Sysplex DASD information is not available on CUA.

- OMEGAVIEW

  OMEGAVIEW provides a 3270 single point of control for viewing all IBM Tivoli OMEGAMON XE statuses at a glance. It allows the display of consolidated monitoring data for z/OS environments. The OMEGAVIEW collects status information from OMEGAMON II CUA or classic monitors.

- CANDLE subsystem

  The CANDLE subsystem provides dynamic I/O support for the OMEGAMON systems.

- Additional monitoring features are available in the z/OS area that may exist for one or more products, such as:
  - EPILOG: historical data collector of OMEGAMON classic information into a set of VSAM database.
  - End To End (ETE) watch: provides end-to-end response time indicator using SNA definite response method. As SNA protocol is no longer widely used, this application is diminished too.
  - CSA analyzer: which is a monitoring helper for CSA systems for IBM Tivoli OMEGAMON XE for z/OS.

- Tivoli Enterprise Monitoring Server or Candle Management Server on z/OS.

  Tivoli Enterprise Monitoring Server acts as the central collector for IBM Tivoli OMEGAMON XE information. In the sysplex environment, each sysplex has one Tivoli Enterprise Monitoring Server that acts as a sysplex proxy. The sysplex proxy, which collects and analyzes sysplex data, can reside in any Tivoli Enterprise Monitoring Server system in the sysplex. You should not run the sysplex proxy in the hub Tivoli Enterprise Monitoring Server.

Most of the IBM Tivoli OMEGAMON XE product comes with a set of started tasks that represent those components. Some of these started tasks are shared between products, and some are specific to a single product, such as the OMEGAMON classic monitor and OMEGAMON II address spaces. Table 2-1 lists an example for IBM Tivoli OMEGAMON XE for z/OS address spaces.

Table 2-1 Product components for IBM Tivoli OMEGAMON XE II for MVS

Component	STC name	Description
IBM Tivoli OMEGAMON XE for MVS classic interface	CANSM2RC	Provides real-time information about an MVS system using the original IBM Tivoli OMEGAMON XE command-driven interface
CUA interface - IBM Tivoli OMEGAMON XE II	CANSM2	Provides real-time information about an MVS system using a menu-driven GUI
Candle Subsystem	CANSCN	Provides dynamic I/O information to IBM Tivoli OMEGAMON XE II

Component	STC name	Description
CSA Analyzer	CANSM2CS	Provides common storage usage information through the IBM Tivoli OMEGAMON XE II for MVS CSAA and CSAF commands
Tivoli Enterprise Monitoring Server	CANSDSST	Collects workload information for IBM Tivoli OMEGAMON XE II
Historical components - EPILOG	CANSM2EZ	VTAM interface to historical collection, provides the EPILOG zoom function.
Historical components - EPILOG	CANSM2HD	CUA component for EPILOG zoom function
End-to-end response time	CANSETZ	Provides IBM Tivoli OMEGAMON XE II with response data

2.1.2 IBM Tivoli Monitoring Services components

This section discusses some components of IBM Tivoli Monitoring Services. This relates to the Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, and Tivoli Enterprise Portal.

Tivoli Enterprise Monitoring Server

The Tivoli Enterprise Monitoring Server is formerly called Candle Management Server or CMS. Tivoli Enterprise Monitoring Server performs the following functions:

- Retrieves data from the monitoring agents and delivers data to the Tivoli Enterprise Portal Server or Candle Management Workstation.
- Sends alerts to the Tivoli Enterprise Portal Server when conditions that are specified in situations are met.
- Receives commands from the Tivoli Enterprise Portal and passes them to the appropriate monitoring agents.
- Optionally, provides a repository for short-term historical data.

Depending on the complexity of your environment and the number of agents you install, Tivoli Enterprise Monitoring Server can be deployed in a hierarchical configuration, where one Tivoli Enterprise Monitoring Server is designated as the *hub* Tivoli Enterprise Monitoring Server and the other Tivoli Enterprise Monitoring Servers as *remote* Tivoli Enterprise Monitoring Servers.

A *hub* Tivoli Enterprise Monitoring Server is the focal point for managing your environment. There can be only one hub Tivoli Enterprise Monitoring Server. It communicates with the Tivoli Enterprise Portal Server, with monitoring agents, and optionally with Tivoli Enterprise Monitoring Servers running as remote in a hierarchical configuration.

A *remote* Tivoli Enterprise Monitoring Server is deployed to distribute the workload of the hub Tivoli Enterprise Monitoring Server. For z/OS monitors, some monitors have to run on a Tivoli Enterprise Monitoring Server; therefore, you have to run a remote Tivoli Enterprise Monitoring Server for every partition for collecting IBM Tivoli OMEGAMON XE for z/OS information. Depending on the complexity of your environment, the number of agents you install, and the number of situations that you monitor, you might need one or more remote Tivoli Enterprise Monitoring Servers. Each remote Tivoli Enterprise Monitoring Server must reside on its own system or workstation and have a unique Tivoli Enterprise Monitoring Server name (node). A remote Tivoli Enterprise Monitoring Server communicates with the hub Tivoli Enterprise Monitoring Server and with monitoring agents running on the same or different systems.

Tivoli Enterprise Portal Server

The Tivoli Enterprise Portal Server (the former CandleNet Portal® Server or CNPS), is a Java application server that enables retrieval, manipulation, and analysis of data from agents. The Tivoli Enterprise Portal Server holds all of the information needed to format the workspaces viewed in the Tivoli Enterprise Portal clients.

The Tivoli Enterprise Portal Server connects the Tivoli Enterprise Portal client to Tivoli Enterprise Monitoring Server. It acts as a server for Tivoli Enterprise Portal and as a client for Tivoli Enterprise Monitoring Server.

The Tivoli Enterprise Portal Server is a collection of software services for the client that enables retrieval, manipulation, and analysis of data from the Candle agent, OMEGAMON agents, or IBM Tivoli Monitoring agents on your enterprise. The Tivoli Enterprise Portal Server connects to the Tivoli Enterprise Monitoring Server. The Tivoli Enterprise Monitoring Server acts as a collection and control point for alerts received from the agents, and collects performance and availability data from the agents. The main, or hub, Tivoli Enterprise Monitoring Server correlates the monitoring data collected by agents and remote servers and passes it to the Tivoli Enterprise Portal Server for presentation and your evaluation.

Tivoli Enterprise Portal Server maps requests from Tivoli Enterprise Portal as queries and sends them to Tivoli Enterprise Monitoring Server. It receives the data returned from Tivoli Enterprise Monitoring Server and processes it to build the view on styles saved in each view’s properties and return it to Tivoli Enterprise Portal.

Tivoli Enterprise Portal

The Tivoli Enterprise Portal, formerly called CandleNet Portal (CNP), is the user interface into the IBM Tivoli OMEGAMON XE product set. It provides a view of your enterprise from which you can drill down to more closely examine components of your system environment. Its application window consists of a Navigator that displays all systems in your enterprise where monitoring agents are installed, and workspaces that include table and chart views of system and application conditions. A true or raised situation causes event indicators to appear in the navigator tree.

Tivoli Enterprise Portal offers two modes of operation: desktop and browser. In desktop mode, the application software is installed on your system. In browser mode, the software is downloaded to your system when you log on to Tivoli Enterprise Portal Server and stored in the Internet Explorer cache. The Java applet code of Tivoli Enterprise Portal is downloaded from the embedded Web server in Tivoli Enterprise Portal Server and run on your desktop machine, thus saving the trouble of installing and maintaining Tivoli Enterprise Portal on each workstation that Tivoli Enterprise Portal desktop mode brings. The URL for the browser mode client is:

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

Monitoring agents

Monitoring agents monitor and collect performance data from a managed system. The agents are installed on the systems or subsystems you want to monitor and communicate with a single Tivoli Enterprise Monitoring Server (remote or hub). They provide data and performance information to the Tivoli Enterprise Monitoring Server and receive instructions from the Tivoli Enterprise Monitoring Server. They are also able to issue commands to the system or application you are monitoring.

These monitoring agents have several names, such as: Tivoli Enterprise Monitoring Agent, OMEGAMON monitoring agent (OMA), and Intelligent Remote Agent (IRA). The monitoring agent can reside on z/OS systems or distributed systems. The architecture and components of z/OS-based monitoring agents are discussed in 2.1.3, "OMEGAMON components on z/OS" on page 26. The distributed monitoring agent is discussed in *Getting Started with IBM Tivoli Monitoring Version 6.1*, SG24-7143.

2.1.3 OMEGAMON components on z/OS

This is the principal relationship between components on z/OS:

- OMEGAMON classic, IBM Tivoli OMEGAMON XE II, and OMEGAVIEW are components that encapsulate each other and provide a degree of abstraction that enables easier user interface with a 3270-based interface. Figure 2-3 shows overall connectivity. The user can connect with a 3270 terminal to either the OMEGAVIEW, OMEGAMON II, or OMEGAMON classic interface.

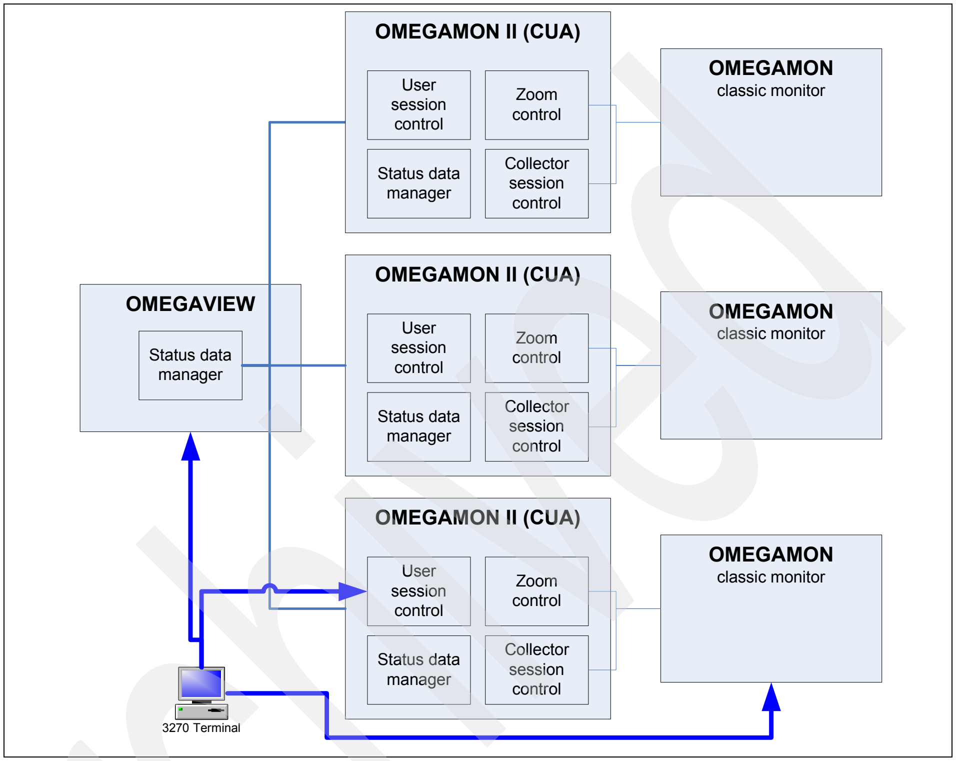


Figure 2-3 IBM Tivoli OMEGAMON XE II vs. IBM Tivoli OMEGAMON XE Classic

An internal VTAM session is set up between the OMEGAMON II CUA interface and OMEGAMON classic interface. The CUA component keeps track of all of the thresholds set in the classic component, and reflects status with status lights using input from Status Data Manager. With multi-session VTAM support, OMEGAVIEW provides navigation into the various OMEGAMON II products it controls.

Internally, OMEGAVIEW Session Manager connects using LU2 to OMEGAMON II, which connects back to the SDM using LU6.2 APPC and adds or modifies status records.

- Sysplex proxy

  To collect and analyze sysplex data, the data elements that are shared among z/OS images in a sysplex environment, one Tivoli Enterprise Monitoring Server must be designated as primary sysplex proxy. The other Tivoli

Enterprise Monitoring Managers are subsequently backup sysplex proxies unless you exclude them from sysplex proxy eligibility. If the primary proxy becomes unavailable, the proxy function is rolled over to the first available backup. The primary sysplex proxy consolidates data from Tivoli Enterprise Monitoring Servers on every image in the sysplex such as coupling facility resource utilization, global enqueue status, XCF performance, and so on. It evaluates the sysplex monitored situations and collects sysplex-level historical data.

Like the hub Tivoli Enterprise Monitoring Server, the proxy system is quite busy. If possible, the sysplex proxy and hub Tivoli Enterprise Monitoring Server should not be on the same system.

Figure 2-4 shows the deployment of a sysplex proxy in a four-member sysplex environment. The primary sysplex proxy is deployed on a host D that is different from the host A where the hub Tivoli Enterprise Monitoring Server is deployed. The dotted line shows the movement of sysplex data from sysplex members A, B, and C to primary sysplex proxy D. The solid line shows the moving of LPAR data of B, C, D, and consolidated sysplex data from the primary sysplex proxy on D to the hub Tivoli Enterprise Monitoring Server.

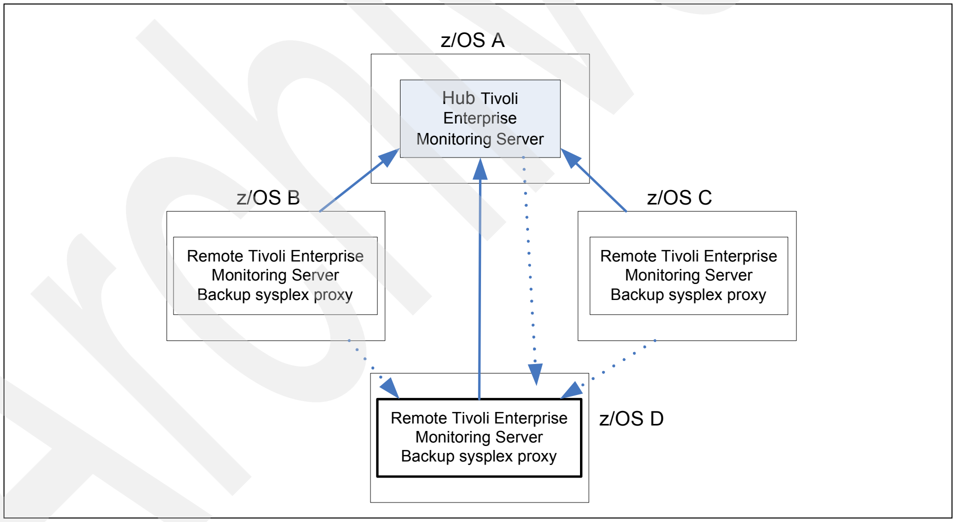


Figure 2-4 Sysplex proxy in sysplex

In 2.3.3, "Failover of sysplex proxy" on page 36, we discuss the failover of sysplex proxy in detail.

Candle Subsystem

Candle Subsystem is a z/OS subsystem running in its own address space that provides dynamic I/O activity to OMEGAMON II products. This subsystem is typically called CNDL.

ITMS:Engine

The ITMS:Engine provides common functions on z/OS for the Tivoli Enterprise Monitoring Server, monitoring agents, and OMEGAMON II components. It provides communications between the following pairs of components:

- Tivoli Enterprise Monitoring Server and monitoring agents
- The hub Tivoli Enterprise Monitoring Server and remote Tivoli Enterprise Monitoring Server
- The hub Tivoli Enterprise Monitoring Server and Tivoli Enterprise Portal Server

It also provides the user and VTAM interfaces for OMEGAMON II.

2.2 Data collection process and mechanism

In Tivoli Enterprise Portal we use two types of information: the view data and enterprise events. The main difference between them is that view data is retrieved only when we access or refresh a workspace, and event data is sent to Tivoli Enterprise Monitoring Server whenever an event occurs.

In this section, we discuss the data collection process for these two types and elaborate on the collection mechanism:

- 2.2.1, “Basic terms” on page 29
- 2.2.2, “View data collection process” on page 30
- 2.2.3, “Event data collection process” on page 31
- 2.2.4, “Situation synchronization mechanism” on page 31
- 2.2.5, “UADVISOR collection mechanism” on page 32

2.2.1 Basic terms

Data collection uses the following basic terms:

- Attribute

  An attribute is a system or application element being monitored by the IBM Tivoli OMEGAMON XE agent, such as Disk Name and Disk Read/Writes per Second. An attribute is sometimes referred to as a column or a field in an

ODBC-compliant database schema used in queries. All agents return their values as attributes. When you open the view or start the situation, Tivoli Enterprise Portal retrieves data samples of the selected attributes.

- Attribute group

  A set of related attributes that can be combined in a data view or a situation. Each type of agent has a set of attribute groups. The attribute group is also considered as a table for queries.

- Query

  Queries are an expression that determine what information is requested from an IBM Tivoli OMEGAMON XE monitoring agent. It is also called pre-filtering to distinguish it from post-filtering or view filters, which fine-tune the data after it has been retrieved by the query. Therefore, query helps in filtering unwanted data and reducing network traffic or processing demands, whereas post-filtering does not.

  A query is assigned to each view within a workspace and is used to request data from the agent every time the workspace is accessed. Multiple views can be associated with the same query.

  The query expression defines the data source, source tables, attributes, and filtering conditions. Chart and table views have a Query tab in their properties page; on this tab you can select which attribute group to gather data from.

  IBM Tivoli OMEGAMON DE enables you to display information from multiple nodes or multiple applications within one workspace; for example, collecting data from three different CICS regions and a DB2 database.

- Situation

  Situations are conditional expressions that compare current attribute values with thresholds. When those thresholds are exceeded and the condition is met, an event shows up in Tivoli Enterprise Portal with the severity or color that was specified for it. For example, if USED_DISK_SPACE is greater than 90%, it shows critical event, which is red. IBM Tivoli OMEGAMON XE is shipped with default product-provided situations with each application.

  You should not modify product-provided situations because if you do (for example, if you change the thresholds), it will be replaced with the default the next time you upgrade or reseed the Tivoli Enterprise Monitoring Server.

2.2.2 View data collection process

View data collection is also called on-demand sampling, because it occurs when a user requests data. Every time a user opens a workspace or every time that workspace is refreshed, the Tivoli Enterprise Portal Server requests a data sample to Tivoli Enterprise Monitoring Server as specified in each view's query.

Based on the query expression that defines what data source and what table to get data from, Tivoli Enterprise Monitoring Server forwards the request to the appropriate agent. The agent is queried for all data that fits the expression and sends results to Tivoli Enterprise Monitoring Server, which then sends it back to Tivoli Enterprise Portal Server. Tivoli Enterprise Portal Server builds the view based on the returned data with the applied filter criteria and styles settings saved in the view properties and returns the view to Tivoli Enterprise Portal for display.

2.2.3 Event data collection process

Event data collection is also called steady-state sampling, because it is driven by interval, not user request. Therefore, even when no users are signed on, the event data collection process still occurs.

Compared with the view data collection process, which is based on the queries embedded in workspace views, the event data collection process is based on situations that detect problems proactively by comparing actual system values with conditions. Situations are stored in the situation table RKDSSITF VSAM file.

All situations are driven from the hub Tivoli Enterprise Monitoring Server. Product-provided situations are distributed to the managed system distribution list. Several distribution lists are created depending on the products that you install, such as *MVS_SYSPLEX for sysplex monitoring, and *HUB for hub offline situations. Customized situations are similarly distributed to the managed system list according to the definition of situations.

The core component in Tivoli Enterprise Monitoring Server that manages situations is called situation monitor or Sitmon. It compiles situations into SQL statements and periodically submits them to the data server. The interval of submission is determined by the sampling interval definition of situations.

CT Data Server (CTDS) is the SQL engine that processes SQL statements for retrieving attributes. Using the Catalog that records the information of column definitions, probe names, and index information, it finds the probe that supports the SQL request. Data server schedules the driving of the monitoring agent and gathers the result rows from the agent. Then it implements sorting and filtering on the result rows specified in the SQL and returns the result rows to SITMON, which computes the situation and possibly generates an event.

2.2.4 Situation synchronization mechanism

In the classic IBM Tivoli OMEGAMON XE, you are allowed to set an interval for every threshold. IBM Tivoli OMEGAMON XE II consolidated these thresholds into lights and typically provided one interval for a group of thresholds. With Tivoli

Enterprise Portal, you can set different intervals for every situation. However, when the interval expires, a data sampling is issued. Therefore many different intervals may cause additional processing.

To reduce the sampling overhead brought by the multiple different intervals, Tivoli Enterprise Monitoring Server uses an optimization method called *situation synchronization*, also known as duper or duperization, to group situation sampling requests and sample the same table and its attribute at the same time. Therefore only one sampling is performed for a set of situations that sample the same attributes at the same interval.

The situation synchronization concept is widely used in the product-supplied situations, so when changing the interval of one product-supplied situation, you may introduce a new sampling rate and double the processing requirement. This additional processing can be significant depending on the number of the started situations for an attribute group.

For customized situations, we recommend that you apply situation synchronization concepts as well. The detailed recipe is covered in 5.2, “Tivoli Enterprise Monitoring Server performance” on page 143 and 5.4, “Tuning situations” on page 162.

2.2.5 UADVISOR collection mechanism

UADVISOR is a special form of situation that is created inside the product and not visible to Tivoli Enterprise Portal. UADVISORs are delivered with the product seed data and installed on all Tivoli Enterprise Monitoring Server images. You cannot modify them.

UADVISOR is widely used for background data collection by WorkLoad Manager (WLM), DB2 Object Analysis, and WebSphere MQ. It is also used by sysplex-related products such as IBM Tivoli OMEGAMON XE on z/OS to gather data from multiple sysplex images.

UADVISOR situations are driven by Sitmon. Similar to any situation, Sitmon compiles it into SQL statements to be submitted to the CT Data Server. Data Server processes the SQL statements and finds the UADVISOR probe that supports the SQL request using the catalog. The UADVISOR probe gathers data from the managed system and supplies the data to the hub Tivoli Enterprise Monitoring Server or sysplex proxy (for sysplex information) to build an in-storage table that is known as VTABLE. Other situations can just evaluate the data at the VTABLE, thus reducing the overhead of driving collection themselves.

Part of the UADVISOR definition is the time interval that they are run on. This interval varies depending on monitoring requirements. Enqueue checking is performed every minute, WLM and DASD are run every five minutes, and others run every three minutes. Sitmon synchronize their launch to a specific time on the sysplex clock. This means that a given UADVISOR will fire in each MVS image at very much the same point in time.

2.3 High availability and failover

In the IBM Tivoli OMEGAMON XE platform architecture on z/OS, the high-availability solution is very important. Some resources reside only on a single system for the whole sysplex environment. In this section, we discuss some high-availability and failover options for the hub Tivoli Enterprise Monitoring Server and sysplex proxy.

The hub Tivoli Enterprise Monitoring Server role is critical as the central repository for all monitoring and event management data, for driving the situations and for running query requests to the appropriate agents. As a sysplex data consolidation point, a sysplex proxy also plays a key role for collecting sysplex data for IBM Tivoli OMEGAMON XE. There is a business need to have Tivoli Enterprise Monitoring Server and sysplex proxy remain operational 24 hours a day and 7 days a week. To deliver nearly non-stop operation, we must ensure their high availability. This is achieved by implementing failover.

In this section, we discuss:

- 2.3.1, "The moveable hub Tivoli Enterprise Monitoring Server" on page 33
- 2.3.2, "Hot standby for distributed system hub" on page 34
- 2.3.3, "Failover of sysplex proxy" on page 36

2.3.1 The moveable hub Tivoli Enterprise Monitoring Server

The moveable hub Tivoli Enterprise Monitoring Server is a procedure that enables creation of an additional Tivoli Enterprise Monitoring Server on z/OS to perform as a hub. The moveable hub implementation enables the hub Tivoli Enterprise Monitoring Server to be started on a different member of the sysplex and allows the Tivoli Enterprise Portal Server to connect transparently whenever the z/OS image starts the Tivoli Enterprise Monitoring Server.

Figure 2-5 on page 34 shows the moveable hub concept.

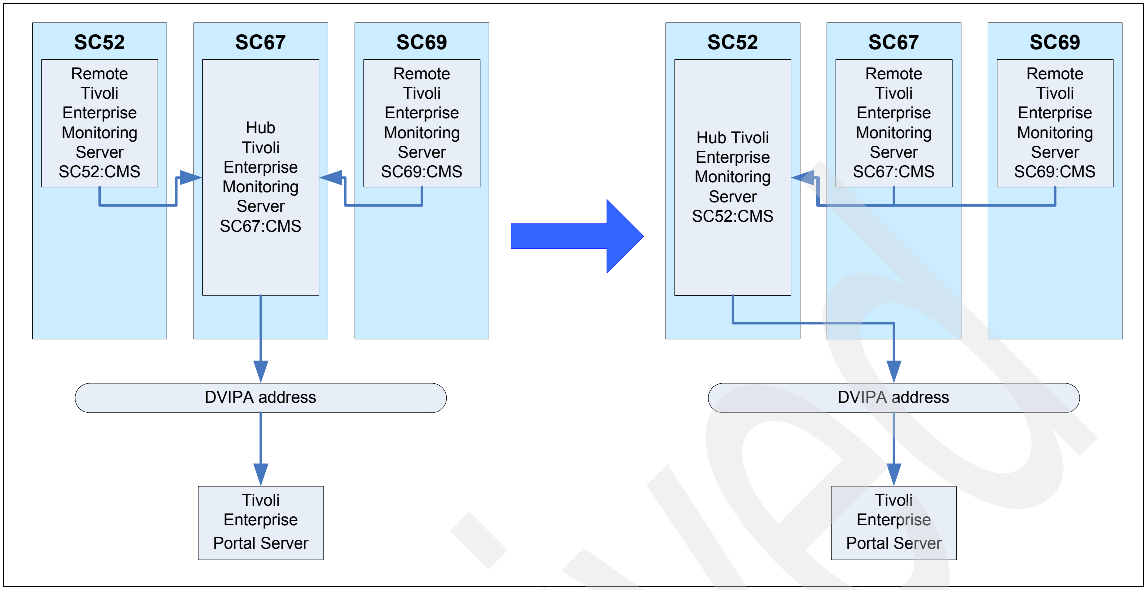


Figure 2-5 The moveable hub

The moveable hub implementation fulfills the need to get the hub Tivoli Enterprise Monitoring Server process to be independent of the sysplex image. The implementation requires strict change management for new products and maintenance. As the hub Tivoli Enterprise Monitoring Server is implemented on a separate runtime environment, any update to the products for the primary runtime environment must also be applied to the hub runtime environment. This is typically related to the seeding of the Tivoli Enterprise Monitoring Server.

Moveable hub typically requires the use of Dynamic Virtual IP Addressing (DVIPA). DVIPA enables the hub to acquire the same IP address, which is different from the physical IP address of the system they are running on.

See 3.5, “Moveable hub implementation” on page 90.

2.3.2 Hot standby for distributed system hub

The high-availability solution for a distributed system hub Tivoli Enterprise Monitoring Server is called a *hot standby* solution. Hot standby Tivoli Enterprise Monitoring Server is a Tivoli Enterprise Monitoring Server, which we designate as a backup, that runs separately from the hub Tivoli Enterprise Monitoring Server at a different location and receives all updates by mirroring the hub Tivoli Enterprise Monitoring Server. It functions to take over the task of the hub Tivoli

Enterprise Monitoring Server when it fails. When the original failing hub is restored its connectivity, it becomes the acting hot standby.

The hot standby designation is performed on the Candle Management Server. It is specified on the Candle Management Workstation property page (Figure 2-6).

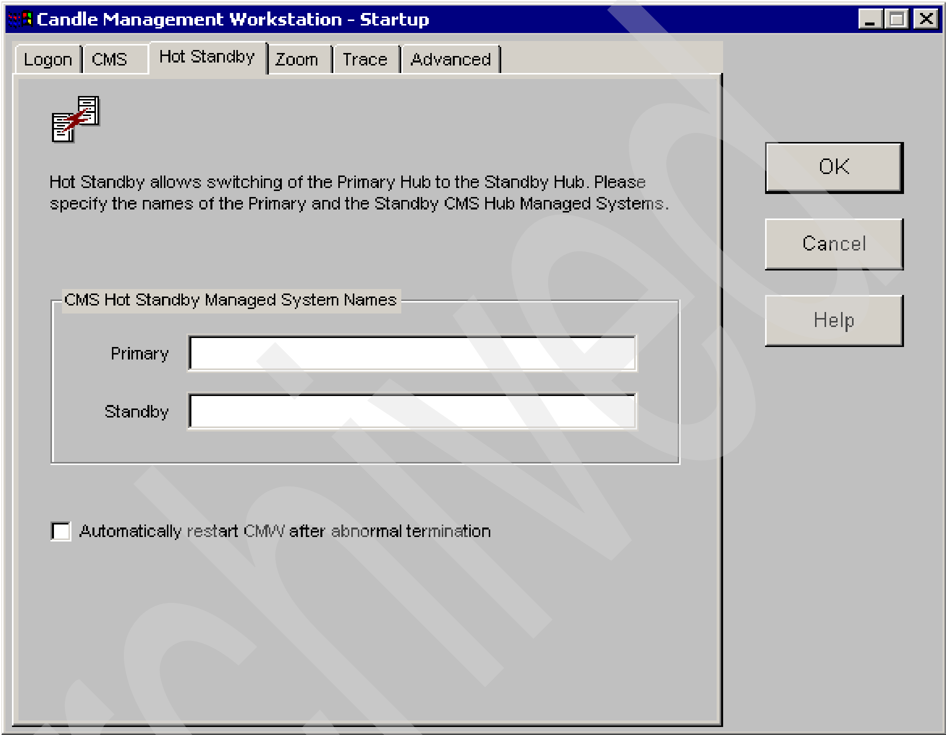


Figure 2-6 Hot standby setup from CMW

Based on the image in Figure 2-6:

- In the Primary field, enter the name of the hub Tivoli Enterprise Monitoring Server to which this CMW is connected.
- In the Secondary field, enter the name of the Tivoli Enterprise Monitoring Server that will assume the functions of the primary Tivoli Enterprise Monitoring Server if the primary Tivoli Enterprise Monitoring Server fails.

The value entered for the Tivoli Enterprise Monitoring Server name on this dialog should be identical to the name specified for the Tivoli Enterprise Monitoring Server's name during installation. The name is case sensitive.

To have the CMW restart automatically after a takeover and connect to the secondary Tivoli Enterprise Monitoring Server, check the box at the bottom of the window.

If your hub Tivoli Enterprise Monitoring Server fails and operations are switched to a backup Tivoli Enterprise Monitoring Server, the persistence count for an event persistence situation is not carried over to the backup Tivoli Enterprise Monitoring Server. Thus, for example, if you create a situation with an event persistence count of 5 and the situation has raised for two intervals, if your primary hub fails, the situation may be restarted or the counters may be reset at the agents that are connected to the hub.

Application tables that are not a part of the basic CT framework (such as MQ configuration product tables) must be copied to the backup Tivoli Enterprise Monitoring Server periodically to prepare for the case where the primary Tivoli Enterprise Monitoring Server hub may fail. A table is eligible to be migrated only if its associated agent has enabled this hot standby feature. It is your responsibility to determine how often you want to perform this migration.

Note: You still need to switch your Tivoli Enterprise Portal Server to connect to the new hub manually.

2.3.3 Failover of sysplex proxy

In this section, we discuss the take over mechanism of the backup sysplex proxy candidate when the primary sysplex fails.

The Tivoli Enterprise Monitoring Server in the first runtime environment (RTE) that you assign to a sysplex during configuration is designated as the primary proxy, and servers that you assign subsequently in the RTEs are defined as backup candidates. You can explicitly exclude any Tivoli Enterprise Monitoring Server from proxy eligibility.

If the primary proxy becomes unavailable, the proxy function is rolled over to the first available backup. When a Tivoli Enterprise Monitoring Server is started, if it is a backup candidate it will wait for up to two minutes to attempt to acquire an enqueue by the name of KLVGLOCK:*sysplex name*. If the Tivoli Enterprise Monitoring Server is the primary candidate it will immediately attempt to get the enqueue. The Tivoli Enterprise Monitoring Server that wins ownership of that enqueue will be the sysplex proxy. The other Tivoli Enterprise Monitoring Servers have an outstanding request for the enqueue and are stacked in the sequence of the enqueue requests as waiters. These run as separate threads in the Tivoli Enterprise Monitoring Server so no vital function is actually waiting. If the current sysplex proxy fails, it will relinquish control of the KLVGLOCK enqueue. The next in line will acquire the enqueue and become the active sysplex proxy.

The new sysplex proxy Tivoli Enterprise Monitoring Server will activate its XCF receiver and will therefore receive UADVISOR data from the other Tivoli Enterprise Monitoring Server in the sysplex. It will re-register as the sysplex's managed system proxy host. This will cause the hub Tivoli Enterprise Monitoring Server to redistribute the situations to the new sysplex proxy. It will also cause Tivoli Enterprise Monitoring Server to direct data query for this sysplex to the new sysplex proxy. Restart of the failed Tivoli Enterprise Monitoring Server does not reacquire the function as sysplex proxy but joins as a backup candidate.

If the failover of Tivoli Enterprise Monitoring Server is also the hub Tivoli Enterprise Monitoring Server, then the failure would mean a loss of the hub, and that is unrecoverable.

The primary sysplex proxy is the only RTE that can define the persistent datastore files for sysplex-level historical data. In the event of the primary sysplex proxy failure and subsequent recovery, the persistent datastore files will be activated on the new takeover sysplex proxy. Historical reporting can resume immediately.

2.4 Historical data collection

Historical data collection provides the ability to store data collected over time. Collecting historical data makes it possible for you to trend analysis and to evaluate the performance of different systems over a period of time.

For IBM Tivoli OMEGAMON XE on z/OS, when historical data is collected, the data is stored in two different locations (Figure 2-7 on page 38):

- Short-term history (up to 24 hours) on z/OS is stored in the Persistent Data Store (PDS) datasets at the monitoring agent or the Tivoli Enterprise Monitoring Server to which the agent connects.
- Long-term data is uploaded to Tivoli Data Warehouse on an hourly or daily basis.

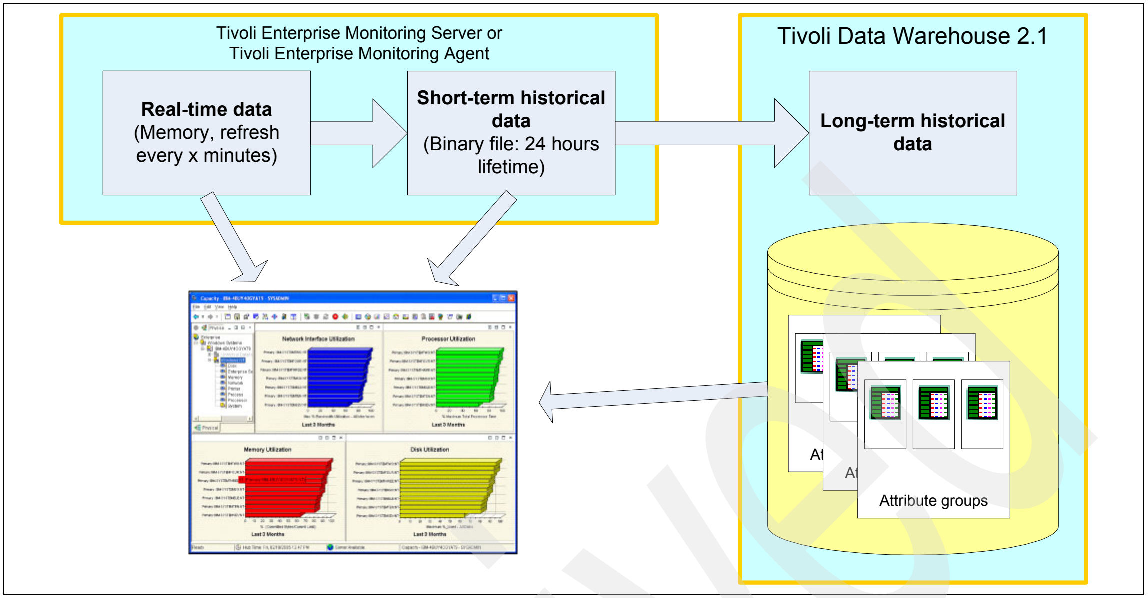


Figure 2-7 Historical data collection

Persistent Data Store (PDS) records and stores 24 hours worth of monitoring data on a z/OS system. You can access it from the Tivoli Enterprise Portal interface or the Candle Management Workstation interface. The data being written to the persistent data store is organized by tables, groups, and datasets. Tivoli Enterprise Monitoring Server provides automatic maintenance for the datasets in the PDS.

Tivoli Data Warehouse provides a medium for long-term historical data store. The Warehouse Proxy agent periodically moves the short-term data from the persistent data store to Tivoli Data Warehouse. It can be configured to warehouse data hourly or daily.

> **Note:** We recommend, if possible, installing warehouse proxy agent on the same machine as the database to save the network traffic from transferring the data twice.

EPILOG provides VSAM-based historical data information from the monitors residing in the OMEGAMON classic or OMEGAMON II address spaces. The information is maintained completely separately from the Tivoli Data Warehouse architecture. This information is also specific to the agent that the EPILOG is attached to and does not provide cross-reporting facility. EPILOG is typically beneficial for system programmers to collect information on specific monitors on z/OS and print out the information periodically.

3

Installation and configuration: IBM Tivoli OMEGAMON XE

This chapter discusses the installation and configuration of IBM Tivoli OMEGAMON XE in our z/OS environment. We also detail the steps that are required to install IBM Tivoli Monitoring Services components. Component installation is discussed for Windows and Linux on zSeries platforms. We implemented scenarios for a z/OS-based Tivoli Enterprise Monitoring Server and a distributed environment Tivoli Enterprise Monitoring Server.

The sections of this chapter are:

3.1 Planning for installation

The successful installation of an enterprise monitoring system takes careful planning. This section provides the steps necessary to plan this installation. Multiple factors must be considered before you begin the installation of the OMEGAMON infrastructure. Each environment has its own prerequisites.

These steps include:

- 3.1.1, “Checking the requirements” on page 40
- 3.1.2, “Implementation options” on page 42
- 3.1.3, “Overview of the installation and configuration process” on page 43

3.1.1 Checking the requirements

For IBM Tivoli OMEGAMON XE, we discuss requirements for hardware, software, and networks.

Hardware and software requirements

The complete system requirements, such as for software level and storage, are supplied in the products Program Directory. Refer to the appropriate Program Directory for the products that you are installing. You should also review the current Preventive Service Planning (PSP) information and acquire all of the necessary maintenance that is available for the product.

The driving system that is used to install IBM Tivoli OMEGAMON XE can run in any hardware environment that supports z/OS V1.4, 5694-A01 or later and IBM System Modification Program/Extended (SMP/E) for z/OS and OS/390 V3.2.0, 5655-G44 or later. Our environment runs z/OS 1.5.

- For Tivoli Enterprise Portal Server: This can run on either a Windows-based system or Linux running on zSeries (with IBM Tivoli Monitoring V6.1). Recommended hardware for a Windows-based system is an Intel® system capable of running Windows XP Pro, Windows 2003, or Windows 2000.

  Software: Windows XP Professional Edition with SP 1 (or higher) or Windows 2000 with SP 3 (or higher).

- Desktop or browser client:

  Tivoli Enterprise Portal desktop client requires Sun™ Java™ V1.3.1 with fixpack 4 or later, or V1.4.2 with fixpack 7 or later.

  Tivoli Enterprise Portal browser client requires Internet Explorer 6 (or higher) with Java Plug-in (same Java release levels as above).

Note: If IBM Tivoli OMEGAMON XE on z/OS is used with the IBM Tivoli OMEGAVIEW product, they should both be installed in the same CSI target and distribution zones. This ensures that the maintenance level of the ITMS:Engine and Tivoli Enterprise Monitoring Server components, which are used by both products, is at the same level. If they are installed in different CSI zones, you should check to ensure the maintenance levels of the ITMS:Engine and Tivoli Enterprise Monitoring Server components in both zones are the same or at a compatible level.

Network requirements

While networking can be considered a hardware component, it does require some attention. The OMEGAMON infrastructure typically uses three communication protocols: IP (UDP), IP.PIPE (TCP), and SNA. These protocols have different requirements in order to properly work with the infrastructure components.

- IP.PIPE specifies the use of the TCP/IP protocol for underlying communications.
- IP specifies the use of the UDP (User Datagram Protocol), which is a packet-based, connectionless protocol.
- SNA specifies the use of Systems Network Architecture (SNA) Advanced Program-To-Program Communications (APPC).

See the program directory and *Getting Started with IBM Tivoli OMEGAMON on z/OS*, SC32-9491, for more details.

To illustrate use of TCP/IP communication with IBM Tivoli OMEGAMON XE on z/OS, Example 3-1 is an excerpt of the command NETSTAT in our environment.

Example 3-1 NETSTAT output

```
NETSTAT
EZZ2350I MVS TCP/IP NETSTAT CS V1R6      TCPIP Name: TCPIP          20:08:58
EZZ2585I User Id  Conn     Local Socket           Foreign Socket         State
EZZ2586I -------  ----     ------------           --------------         -----
EZZ2587I CANSDS52 00007D91 0.0.0.0..2099          0.0.0.0..0             Listen
EZZ2587I CANSDS52 00007D97 0.0.0.0..1918          0.0.0.0..0             Listen
EZZ2587I CANSDS52 00007D94 9.12.4.42..2100        9.12.4.42..1920        Establish
EZZ2587I CANSM2   00007D62 0.0.0.0..1920          0.0.0.0..0             Listen
EZZ2587I CANSM2   00007D95 9.12.4.42..1920        9.12.4.42..2100        Establish
EZZ2587I CANSM2   00007D61 0.0.0.0..2089          0.0.0.0..0             Listen
EZZ2587I CANSM2   00007D64 0.0.0.0..6014          0.0.0.0..0             Listen
EZZ2587I CANSDS52 00007D99 0.0.0.0..1036          *..*                   UDP
EZZ2587I CANSDS52 00007D93 127.0.0.1..1035        *..*                   UDP
```

```
EZZ2587I CANSDS52 00007D9B 0.0.0.0..1037         *..*                   UDP
EZZ2587I CANSDS52 00007D98 0.0.0.0..1918         *..*                   UDP
EZZ2587I CANSDS52 00007D9C 0.0.0.0..1038         *..*                   UDP
EZZ2587I CANSDS52 00007D9D 0.0.0.0..1039         *..*                   UDP
EZZ2587I CANSM2   00007D65 0.0.0.0..1033         *..*                   UDP
EZZ2587I CANSM2   00007D63 127.0.0.1..1032       *..*                   UDP
EZZ2587I CANSM2   00007D67 0.0.0.0..1034         *..*                   UDP
```

3.1.2 Implementation options

Depending on your existing configuration and future needs, there are several different paths that you might be able to take to implement IBM Tivoli OMEGAMON XE. This section provides a general road map for achieving this. The primary consideration are:

- The existence of a previous version of OMEGAMON

  When you already have a previous installation of OMEGAMON, you can choose whether to install IBM Tivoli OMEGAMON XE V3.1 over the existing installation or install the new product separately.

  Installing in the same CSI will replace the common components of the product and you can upgrade your installation according to the migration scenario in the product manual. However, this approach requires that you have a sound backup to allow a fallback. As our environment does not have a previous version of OMEGAMON, we are not testing this upgrade scenario.

  Installing on a different CSI or by creating a different installation zone enables you to have both products running side-by-side until you decide to use the new product. However, this scenario may require you to manually migrate some of the customization that you have already performed in the older product. We demonstrate a fresh installation in 3.2, "SMP/E environment" on page 46 and 3.3, "Configuring IBM Tivoli OMEGAMON XE" on page 50.

  Note: Refer to Tivoli zone creation samples online at:

  `http://www.ibm.com/support/docview.wss?rs=660&context=SSZJDU&uid=swg21066230&loc=en_US&cs=utf-8&lang=en`

- Using a down-level monitor

  Although IBM Tivoli OMEGAMON XE V3.1 comes with most of the major monitoring suite, you still might need to use a down-level monitor that does not have a replacement in V3.1, such as the UNIX System Services monitor. You can use the new implemented infrastructure with this existing monitor. If you migrate from a previous version, then all of the customization dialogs are already aware of this existing product.

However, if you install on a new CSI or perform a fresh install of the product, you must seed the Tivoli Enterprise Monitoring Server. See 3.4.4, “Seeding our z/OS hub Tivoli Enterprise Monitoring Server” on page 85.

- Plan on integration to IBM Tivoli Monitoring V6.1

  Integration of IBM Tivoli OMEGAMON XE V3.1 monitors and some of the older OMEGAMON versions into IBM Tivoli Monitoring V6.1 is supported. We explain this and some considerations in 3.6, “Using IBM Tivoli Monitoring V6.1” on page 100. We use the Linux on zSeries platform to illustrate the pure zSeries solution with IBM Tivoli Monitoring V6.1 as the IBM Tivoli Monitoring Services.

> **Note:** All IBM Tivoli OMEGAMON XE Version 3.1 products are packaged with Tivoli Enterprise Monitoring Server V360 and Tivoli Enterprise Portal Server V196. The IBM Tivoli OMEGAMON XE V3.1. products can run with IBM Tivoli Monitoring V6.1 product number 5724-C04 and IBM Tivoli Monitoring Services on z/OS V6.1, product number 5698-A79. If you are currently running IBM Tivoli OMEGAMON XE V3.1 products and you want to upgrade to IBM Tivoli Monitoring V6.1 or IBM Tivoli Monitoring Services on z/OS V6.1, or if you want to run your IBM Tivoli OMEGAMON XE V3.1 product in an environment where Tivoli Monitoring Services V6.1 is already installed, refer to the document “OMEGAMON V350/360 agent interoperability,” available by searching at:
>
> http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/index.jsp?toc=/com.ibm.itm.doc/toc.xml

3.1.3 Overview of the installation and configuration process

In Version 3.1, IBM Tivoli OMEGAMON XE is delivered using the standard IBM enterprise software fulfillment processes by either electronic or physical delivery. Thus installation and configuration is performed using this process:

- The standard SMP/E processing RECEIVE - APPLY - ACCEPT method that creates the target and distribution libraries, unless you are installing the product from ServerPac or CBPDO, in which the SMP/E processing has been performed by IBM.
- Configuration of the IBM Tivoli OMEGAMON XE product using the IBM Configuration Tool, formerly known as CICAT. IBM Configuration Tool is an ISPF-based menu system that enables easy configuration of the OMEGAMON product. The menu system can sometimes go quite deep (deep enough that we may get lost). IBM Configuration Tool performs these tasks:
  - Defining and building the runtime environment (RTE) that will be used to run the product
  - Configuring system parameters used for the product components

These processes are performed individually for each monitoring agent product to be installed. We go through the Configuration Tool in 3.3, “Configuring IBM Tivoli OMEGAMON XE” on page 50.

- Completing the configuration by setting up the needed information on the IBM Tivoli Monitoring Services:
  - Before you can use an IBM Tivoli OMEGAMON XE on z/OS monitoring agent, the remote Tivoli Enterprise Monitoring Server that the agent reports to and the hub Tivoli Enterprise Monitoring Server that the remote server reports to must be initialized with application data. This process is known as *seeding*. Seeding adds product-provided situations, templates, Enterprise Information Base (EIB) tables, and the product-specific data to Tivoli Enterprise Monitoring Server. Remote Tivoli Enterprise Monitoring Servers belonging to an RTE are seeded within the configuration tool, but the hub Tivoli Enterprise Monitoring Server must be seeded from the product CD-ROM.

    Hubs on UNIX and Windows systems are seeded locally; hubs on z/OS are seeded from the Windows system that hosts the Tivoli Enterprise Portal Server.
  - Tivoli Enterprise Portal Server presentation rebuilding for the additional queries and workspaces that come with the monitoring agent.
  - Tivoli Enterprise Portal client modification for the additional help files and display resources for the monitoring agent.

3.1.4 Assigning and designing the runtime environment

The runtime environment (RTE) is a set of z/OS runtime libraries and VSAM datasets required for the execution of IBM Tivoli OMEGAMON XE products on a z/OS system image. There are three different types of runtime environment:

FULL Allocates both private and base libraries in a single set of datasets. This type of RTE is typically fit for a single system image with only a single product.

BASE Allocates base libraries only; the base RTE will be shared by all participating images. A base RTE must be used in conjunction with SHARING RTEs.

SHARING Allocates private libraries only. Sharing RTEs share base libraries with a base RTE. You can also use a FULL RTE for a base dataset. Typically you define a separate sharing RTE for each z/OS sysplex member. SHARING RTE must be used in conjunction with either BASE or FULL RTE. The sharing RTE consists of image-specific libraries that cannot be shared.

The runtime environment datasets are known to start with the prefix R. These are some sample datasets that are part of a runtime environment:

- Base RTE datasets

RKANMOD	Load module, must be authorized
RKANMODL	Load module, must be authorized
RKANSAS	Programs and macros
RKANSQL	SQL and data members to populate TEMS
RKANDATR	Attributes and attribute groups definitions
RKANCLI	ISPF clists
RKANEXEC	REXX programs
RKANISP	ISPF panels
RKANPENU	English US panels
RKANHENU	English US help
RKNSLOCL	Local codepage information
RKxxHELP	Help information
RKxxPROC	OMEGAMON classic functions

- Sharing RTE datasets

RKANPAR	Parameter members
RKANSAM	Sample jobs or started tasks
RKANCMD	Command list for running the OMEGAMON tasks
RKANDATV	Populated internal tables

We discuss RTE creation and configuration in 3.3, “Configuring IBM Tivoli OMEGAMON XE” on page 50. However, as configuration of the RTE affects how the datasets are used in IBM Tivoli OMEGAMON XE, we offer some sample scenarios using the RTEs:

- A single system in the sysplex (a monoplex) does not require a separate base RTE and sharing RTE, which would only introduce overhead in maintenance and operation. However, as you use more IBM Tivoli OMEGAMON XE products, you may want to implement different RTEs for each of them.
- Sharing RTEs with a base RTE library for a standard sysplex is the most common scenario, and we use it in 3.3, “Configuring IBM Tivoli OMEGAMON XE” on page 50. One of the sharing RTEs is designated to be the hub Tivoli Enterprise Monitoring Server, while other members of the sharing RTEs are used as remote Tivoli Enterprise Monitoring Server. Figure 3-1 on page 46 depicts this RTE configuration.

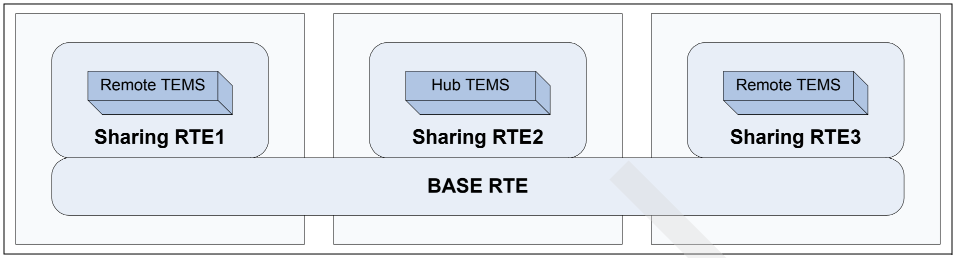


Figure 3-1 Standard base and sharing RTEs

- Another approach that we consider is to have a separate RTE to act as the hub Tivoli Enterprise Monitoring Server, while each sysplex member uses a separate sharing RTE containing the remote Tivoli Enterprise Monitoring Server (Figure 3-2). This configuration is useful for creating a moveable hub Tivoli Enterprise Monitoring Server.

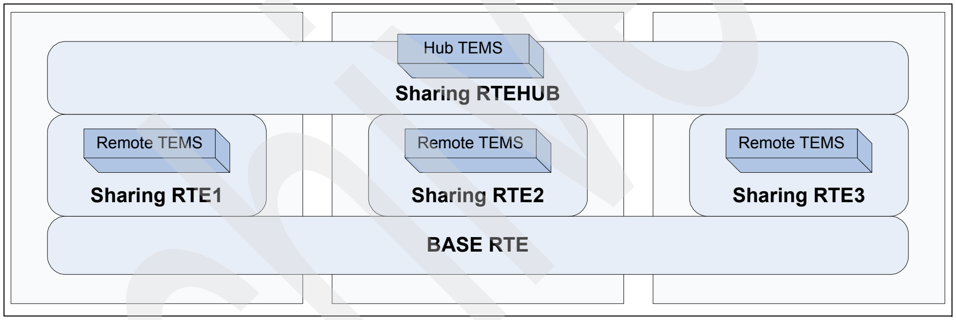


Figure 3-2 Base and sharing RTEs with separate RTE for hub Tivoli Enterprise Monitoring Server

3.2 SMP/E environment

SMP/E can be run using either batch jobs or dialogs under Interactive System Productivity Facility/Program Development Facility (ISPF/PDF). SMP/E dialogs help you interactively query the SMP/E database, as well as create and submit jobs to process SMP/E commands. IBM Tivoli OMEGAMON XE V3.1 supplies program directories to install the product using SMP/E. The program directories that we use are:

- *Program Directory for IBM Tivoli OMEGAMON for IMS on z/OS V3.1.0*, GI11-4077
- *Program Directory for IBM Tivoli OMEGAMON for Mainframe Networks V3.1.0*, GI11-4078

- *Program Directory for IBM Tivoli OMEGAMON for Storage on z/OS V3.1.0*, GI11-4079
- *Program Directory for IBM Tivoli OMEGAMON for CICS on z/OS V3.1.0*, GI11-4080
- *Program Directory for IBM Tivoli OMEGAMON on z/OS V3.1.0*, GI11-4081

Our SMP/E environment includes the products and FMIDs shown in Table 3-1.

Table 3-1 FMID list

Product name	FMID list
IBM Tivoli OMEGAMON XE on z/OS V3.1.0	HKM5310 HKM2550 HKSB550
IBM Tivoli OMEGAMON XE for Mainframe Networks V3.1.0	HKN3310 HKON550
IBM Tivoli OMEGAMON XE for Storage on z/OS V3.1.0	HKS3310 HKDF550 HKSB550
IBM Tivoli OMEGAMON XE for IMS on z/OS V3.1.0	HKI5310 HKI2550
IBM Tivoli OMEGAMON XE for CICS on z/OS V3.1.0	HKC5310 HKC2550
Common components for V3.1.0 products	HKCI310 HKDS360 HKLV190 HKET550[1] HKOB550[2]
IBM Tivoli OMEGAMON XE for OS/390 UNIX System Services V2.2.0	HKOE220
Notes: [1] All except IBM Tivoli OMEGAMON XE for Storage on z/OS [2] All except IBM Tivoli OMEGAMON XE for Mainframe Network	

Table 3-2 on page 48 lists the resulting FMIDs for our IBM Tivoli OMEGAMON XE V3.1. We have all of these FMIDs in a single Consolidated Software Inventory (CSI).

Table 3-2 FMIDs and components

FMID	COMPID	Component name	RETAIN® release
HKCI310	5608A41CC	Configuration Assistance Tool (ICAT)	310
HKDS360	5608A2800	Candle Management Server on z/OS	360
HKET550	5608A4155	EE End-To-End	550
HKLV190	5608A41CE	ITMS:Engine	190
HKOB550	5608A41OB	OMNIMON Base	10B
HKSB550	5608A41SP	Shared Probes	550
HKM5310	5698A5900	OMEGAMON on z/OS	310
HKM2550	5608A0400	OMEGAMON II for MVS	550
HKN3310	5608A4000	OMEGAMON for Mainframe Networks	310
HKON550	5608A3700	OMEGAMON II for Mainframe Networks	550
HKC5310	5698A5800	OMEGAMON for CICS on z/OS	310
HKC2550	5608A0100	OMEGAMON II for CICS	550
HKI5310	5698A3900	OMEGAMON for IMS on z/OS	310
HKDF550	5698A4600	OMEGAMON II for SMS	550
HKS3310	5698A4600	OMEGAMON for Storage	310
HKI2550	5608A0300	OMEGAMON II for IMS	550
AKOE220	5608A2700	OMEGAMON for OS/390 UNIX System Services	220

Some of these FMIDs are shared between products. Apart from these shared FMIDs, some of the target and distribution libraries are also shared, and multiple FMIDs use the same libraries. These shared libraries require you to have a larger library allocation to accommodate all products that use the same libraries.

A typical SMP/E-based installation performs the following steps:

1. Allocate the distribution and target libraries.
2. Define the libraries to SMP/E as DDDEF.
3. Receive the FMID to SMP/E so that SMP/E is aware of its existence and source data.
4. Perform the apply process to populate the target libraries.
5. Perform the accept process to populate the distribution libraries.

Table 3-3 lists the sample jobs that can be used to perform the SMP/E installation for each product.

Table 3-3 Sample jobs

REL file	Allocation	DDDEF	Receive	Apply	Accept
IBM.HKN3310.F8	KN3J0ALO	KN3J0DDF	KN3J0REC	KN3J0APP	KN3J0ACC
IBM.HKM5310.F10	KM5J0ALO	KM5J0DDF	KM5J0REC	KM5J0APP	KM5J0ACC
IBM.HKI5310.F7	KI5J0ALO	KI5J0DDF	KI5J0REC	KI5J0APP	KI5J0ACC
IBM.HKC5310.F8	KC5JALO	KC5JDDF	KC5JREC	KC5JAPP	KC5JACC
IBM.HKS3310.F9	KS3J0ALO	KS3J0DDF	KS3J0REC	KS3J0APP	KS3J0ACC

Because some products use the same dataset libraries, the sample allocation jobs may not be successful for the second product that you install, as the datasets may already exist. You also need to plan the size of allocation carefully as the sample jobs in Table 3-3 accommodate only enough allocation for the respective product. If you try to use the shared libraries for all of the products, the installation will fail for being out of space in these shared libraries. These products may share one or more of these libraries:

- 5698-A57 IBM Tivoli OMEGAMON for WebSphere Business Integration V1.1.0
- 5698-A58 IBM Tivoli OMEGAMON for CICS on z/OS V3.1.0
- 5698-A59 IBM Tivoli OMEGAMON on z/OS V3.1.0
- 5698-A46 IBM Tivoli OMEGAMON for Storage on z/OS V3.1.0
- 5698-A40 IBM Tivoli OMEGAMON for Mainframe Networks V3.1.0
- 5698-A39 IBM Tivoli OMEGAMON for IMS on z/OS V3.1.0
- 5655-P08 IBM Tivoli OMEGAMON for DB2 Performance Monitor on z/OS V3.1.0
- 5655-P07 IBM Tivoli OMEGAMON for DB2 Performance Expert on z/OS V3.1.0
- 5698-A77 IBM Tivoli Composite Application Manager for SOA V6.1.0
- 5698-A71 IBM Tivoli Composite Application Manager for WebSphere V6.0.0
- 5698-A78 IBM OMEGAMON z/OS Management Console V1.1.0
- 5698-A72 IBM Tivoli OMEGAMON DE on z/OS V3.1.0
- 5698-A79 IBM Tivoli Monitoring Services on z/OS V6.1.0

Table 3-4 on page 50 lists all target datasets in our installation, whether they are shared, and the number of tracks that each has. The associated distribution libraries have the same names with a D prefix.

Table 3-4 DDDEF list

Shared DDDEF	# tracks	Non-shared DDDEF	# tracks
TKANCMD	7	TKANCLI	2
TKANCUS	3245	TKANDATR	158
TKANDATV	574	TKANEXEC	9
TKANHENU	1009	TKANHELP	4
TKANMAC	168	TKANISP	8
TKANMOD	766	TKANSAS	345
TKANMODL	1540	TKCIINST	98
TKANMODS	352	TKEPHELP	10
TKANPAR	57	TKNSLOCL	70
TKANPENU	10395	TKOBHELP	30
TKANPKGI	709	TKOCHELP	6
TKANSAM	541	TKOCPROC	102
TKANSQL	50	TKOIHELP	38
		TKOIPROC	123
		TKOMHELP	58
		TKOMPROC	86

3.3 Configuring IBM Tivoli OMEGAMON XE

The IBM Configuration Tool was known as Candle Installation and Configuration Assistance Tool (CICAT). The IBM Configuration Tool is still used to perform both installation and configuration of previous versions of Candle products. In this context, the tool is called the installation and configuration tool. However, for IBM Tivoli OMEGAMON XE V3.1 monitoring agents, the tool is used only for configuration, because these products are installed using SMP/E.

In this section, we walk you through IBM Configuration Tool panels. The process that we explain includes:

- 3.3.1, “IBM Configuration Tool overview” on page 51
- 3.3.2, “Preparing the configuration tool” on page 54
- 3.3.3, “Working with the runtime environment” on page 56

- 3.3.4, “Configuring Tivoli Enterprise Monitoring Server” on page 62
- 3.3.5, “Configuring OMEGAMON II for MVS” on page 68
- 3.3.6, “Configuring IBM Tivoli OMEGAMON XE on z/OS” on page 73

3.3.1 IBM Configuration Tool overview

The IBM Configuration Tool assists you in defining your runtime environment. The processing in the configuration tool creates and customizes all of the runtime datasets required to support the IBM Tivoli OMEGAMON XE. See 3.1.4, “Assigning and designing the runtime environment” on page 44.

The IBM Configuration Tool can be invoked in two ways:

- Interactive mode: A set of ISPF dialogs to collect customization values and execute the creation of the runtime environment.
- Batch mode: A single batch job that enables non-interactive creation and definition of runtime environment; this is useful for replicating the runtime environment to different z/OS systems.

Figure 3-3 summarizes IBM Configuration Tool processing in the interactive mode.

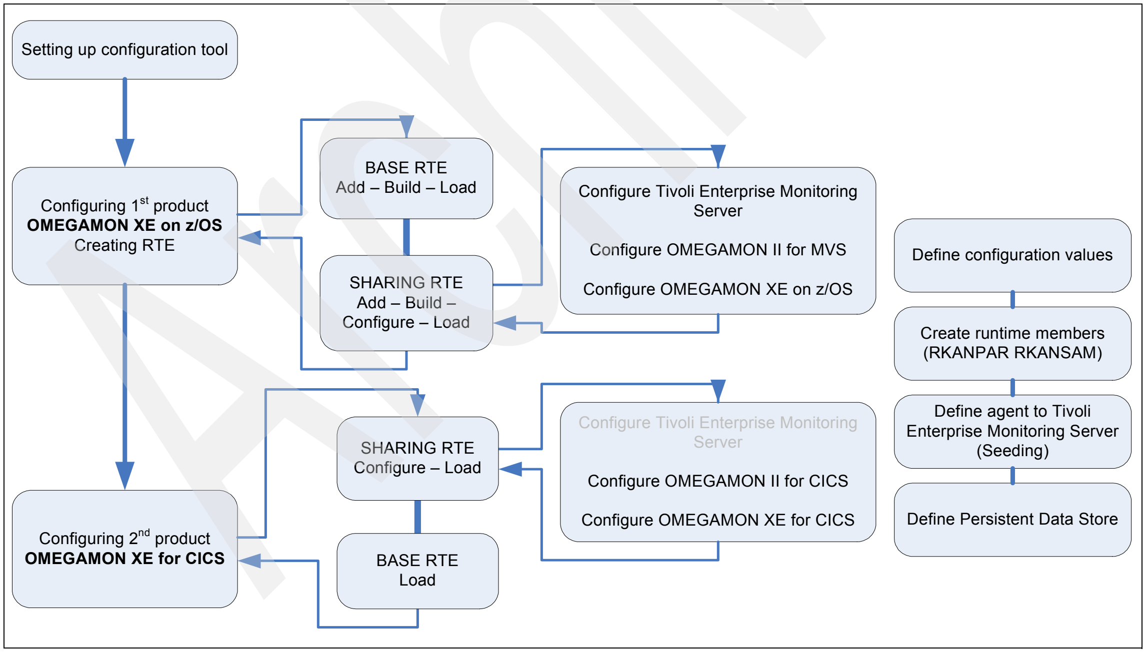


Figure 3-3 Overview of configuration tool processing

Figure 3-3 on page 51 shows a sample sequence of using the IBM Configuration Tool. The tool allows easy configuration of IBM Tivoli OMEGAMON XE products. The configuration steps include configuring the product-specific parameters into the runtime environment. Initially, you must define the appropriate RTEs, as discussed in 3.1.4, “Assigning and designing the runtime environment” on page 44.

You then load the product into the base RTE. For a sharing or a full RTE, you must configure the product parameter first before loading the content of the RTE. Each product consists of several components: for IBM Tivoli OMEGAMON XE V3.1.0, the components typically are Tivoli Enterprise Monitoring Server, OMEGAMON II component, and OMEGAMON monitoring agent. Table 3-5 shows the products that are installed in our environment and the components that they contain.

Table 3-5 Product to components

Product	Component
IBM Tivoli OMEGAMON XE on z/OS	Candle Management Server OMEGAMON II for MVS IBM Tivoli OMEGAMON XE on z/OS
IBM Tivoli OMEGAMON XE for CICS on z/OS	Candle Management Server OMEGAMON II for CICS IBM Tivoli OMEGAMON XE for CICS on z/OS
IBM Tivoli OMEGAMON XE for IMS on z/OS	Candle Management Server OMEGAMON II for IMS IBM Tivoli OMEGAMON XE for IMS on z/OS
IBM Tivoli OMEGAMON XE for Mainframe Network	Candle Management Server OMEGAMON II for Mainframe Networks OMEGAMON for Mainframe Networks
IBM Tivoli OMEGAMON XE for Storage on z/OS	Candle Management Server OMEGAMON II for SMS IBM Tivoli OMEGAMON XE for Storage on z/OS
OMEGAMON for UNIX System Services	Candle Management Server OMEGAMON for OS/390 UNIX System Services

RTE processing consists of the following steps:

Create Defines the RTE and its basic properties, such as dataset names and variables supports. This is performed only once.

Build Generates a JCL to define the datasets for the RTE. This must be performed for each product that defines its own datasets in the RTE.

Configure Interactive dialogs and panels that enable the configuration of RTE components. This is performed for every product for the sharing RTEs and full RTEs.

Load Loads the actual libraries and populates the databases in the RTE with configuration information that has been collected. You should load the libraries after configuring all products.

In the configuration tool, the base RTE is does not have to be configured. However, you need to load the base library every time you configure a product so that the new definitions are loaded to the base library.

Most of the work in the configuration tool resides in configuring the RTE. As shown in Figure 3-3 on page 51, configuration shows the components of a product. You need to configure all components that have not been configured. For each component, the typical configuration steps are:

1. Entering configuration values using a set of ISPF panels. There is extensive help information for each of the fields and panels.
2. Creating the runtime member, basically generating files in RKANPAR and RKANSAM based on the values that you enter from step 1.
3. Defining the agent to Tivoli Enterprise Monitoring Server (only when configuring a Tivoli Enterprise Management Agent). This is similar to the seeding for distributed Tivoli Enterprise Monitoring Server.
4. Defining persistent data store datasets and definitions.

Typically, after you complete the configuration and loading of the RTEs, you need to perform additional tasks that affect z/OS system datasets outside of the RTE. Some examples of these additional tasks are:

- Defining a Candle subsystem
- Authorizing datasets
- Creating the started task procedure
- Defining a VTAM major node

The configuration tool is restartable. If necessary, you can end the dialog, start it again, and continue from the point of interruption. If you have an earlier version of the CICAT on your z/OS system, installing IBM Tivoli OMEGAMON XE V3.1.0 automatically replaces it with IBM Configuration Tool Version 3.1.0 during SMP/E installation.

3.3.2 Preparing the configuration tool

Before you can use the configuration tool, you need to perform several tasks:

1. Your first step after installing the contents of the product tape is to copy the contents of KCIINST into INSTLIB to be able to start the configuration tool. Figure 3-4 shows the JCL that we used.

```
//INSTCOPY JOB ,,CLASS=A,NOTIFY=&SYSUID
//*
//COPY     EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=*
//IN       DD DSN=OMEGAMON.TKCIINST,DISP=SHR
//OUT      DD DSN=OMXE310S.INSTLIB,DISP=(NEW,CATLG),VOL=SER=OMXE3C,
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=8880),SPACE=(CYL,(7,1,88))
//SYSIN    DD *
  C O=OUT,I=((IN,R))
/*
```

Figure 3-4 Create the work library

2. We can now start the configuration tool. From an ISPF TSO session, invoke the newly created library from Figure 3-4. The command is:

   `EXEC 'OMXE310S.INSTLIB'`

   The configuration tool displays the copyright panel as shown in Figure 3-5. You can see that our version of the configuration tool is 310.01.

```
                 Installation and Configuration

                        Assistance Tool

                        Version 310.01



                 (C) Copyright IBM Corp. 1992-2004
             Licensed Material - Program Property of IBM
```

Figure 3-5 Configuration tool copyright screen

3. From the Main Menu (Figure 3-6), initially, we set up our work environment.

```
---------------------------------MAIN MENU--------------------------------
OPTION ===> 1

 Enter the number to select an option:

    1   Set up work environment

    2   Install products or maintenance

    3   Configure products

    I   Installation information
    S   Services and utilities




       Installation and Configuration Assistance Tool Version 310.01
                      (C) Copyright IBM Corp. 1992-2004
                 Licensed Material - Program Property of IBM

 F1=Help  F3=Back
```

Figure 3-6   Configuration tool main menu

4. In the Set Up Work Environment menu, shown in Figure 3-7:
   a. Type 1 (Specify options) to specify allocation and processing values that will be used to create the work data sets that are needed by the Configuration tool.
   b. Type 2 (Allocate work libraries) to allocate the configuration tool work libraries.

```
--------------------------- SET UP WORK ENVIRONMENT ------------------------
OPTION ===>
                                                           Last selected
 Enter the number to select an option:                      Date     Time

    1   Specify options                                    05/11/01  15:57
    2   Allocate work libraries                            05/11/10  11:16









Note: Once you create and submit the "Allocate work libraries" job,
      you must exit the installer and allow the job to run before
      restarting the installer.

 F1=Help  F3=Back
```

Figure 3-7 Set up the work environment

> **Note:** After you create and submit the Allocate work libraries job, you must exit the configuration tool and allow the job to run before starting the configuration tool again.

3.3.3 Working with the runtime environment

When the configuration tool is ready, we use it to define the runtime environment. Back in the Main Menu (Figure 3-6 on page 55), type 3 to configure products. We do not select 2 to install the product as installation is performed using SMP/E outside of the configuration tool. Figure 3-8 on page 57 shows the Configure Products menu. Type 1 to set up the configuration environment.

```
----------------------------- CONFIGURE PRODUCTS---------------------------
OPTION ===> 1

 Enter the number to select an option:

    1   Set up configuration environment

    2   Select product to configure

    I   Configuration information               <=== Revised
    S   Services and utilities

F1=Help  F3=Back
```

Figure 3-8   Configure products menu

Figure 3-9 shows the setup for the configuration environment. We provide the high-level qualifier that we use in our environment.

```
---------------------- SET UP CONFIGURATION ENVIRONMENT--------------------
COMMAND ===>

  *** High-level qualifiers are locked.

 RTE allocation routine   ==> IKJEFT01  (IKJEFT01/IEFBR14)

 Runtime                                         Unit/     Storclas/
 Datasets      High-Level Qualifier              VolSer    Mgmtclas    PDSE
 VSAM          OMEGAMON                          3390
                                                 OMXE3C
 Non-VSAM      OMEGAMON                          3390                  N
                                                 OMXE3C
 Work    ...............................         SYSDA

 SMP/E
 Datasets      High-Level Qualifier
 Target        OMEGAMON
```

Figure 3-9   Configuration environment and high level qualifiers

Back at the Configure Products menu in Figure 3-8, type 2 to choose the product.

The list in Figure 3-10 shows the products that are installed and available to be configured in this system. We use IBM Tivoli OMEGAMON XE on z/OS V3.1.0 as the first product that we configure. Depending on your environment, you may have different lists and choose a different product.

```
---------------------------- PRODUCT SELECTION MENU -------------------------
COMMAND ===>

 Actions: S Select product

    IBM Tivoli OMEGAMON for CICS on z/OS V3.1.0
    IBM Tivoli OMEGAMON for IMS on z/OS V3.1.0
    IBM Tivoli OMEGAMON for Mainframe Networks V3.1.0
    IBM Tivoli OMEGAMON for Storage on z/OS V3.1.0
 s  IBM Tivoli OMEGAMON on z/OS V3.1.0
    OMEGAMON for OS/390 UNIX System Services V220
F1=Help   F3=Back   F5=Refresh   F7=Up   F8=Down
```

Figure 3-10 Product selection menu

As this is the first OMEGAMON product that we configure, we have not defined any runtime environment (RTE), hence we need to define our RTE. Depending on how you would use the RTE, you can either define a full RTE or a set of base RTE and sharing RTEs.

We install OMEGAMON in a sharing RTE as we will install an RTE on each of our LPARs. Our DASD is shared across the sysplex environment. Allocating a sharing RTE defines only private libraries, and will share the base libraries with our base RTE.

1. First we add our base RTE called `BASE`. We type `A` in the Action field as shown in Figure 3-11.

```
------------------------- RUNTIME ENVIRONMENTS (RTEs)-----------------------
COMMAND ===>

Actions: A Add RTE, B Build libraries, C Configure,
         L Load all product libraries after SMP/E,
         D Delete, U Update, V View values, Z Utilities

Action Name     Type    Sharing  Description
  a    BASE     BASE             Base RTE for SC52, SC69, SC67
Enter=Next  F1=Help  F3=Back  F7=Up  F8=Down
```

Figure 3-11 Base RTE creation

2. Figure 3-12 shows the base RTE definition.

```
-------------------------- ADD RUNTIME ENVIRONMENT ------------------------
COMMAND ===>

RTE: BASE Type: BASE    Desc: Base RTE for SC52, SC69, SC67

  Libraries  High-level Qualifier       Volser Unit     Storclas Mgmtclas PDSE
  Non-VSAM   OMEGAMON                   OMXE3C 3390                        N
  VSAM       OMEGAMON                   OMXE3C

  Mid-level qualifier ==>  BASE       (Optional for a base RTE)

  JCL suffix              ==> CANS
  SYSOUT class            ==> X
  Diagnostic SYSOUT class ==> X
  Load optimization       ==> N   (Y, N)
```

Figure 3-12 Base RTE properties

3. Back in the RTE list panel, which is similar to Figure 3-11 on page 58, we build the BASE RTE using the option B. This option generates a batch job that you must submit. Verify that the job completes successfully: All condition codes should be 0000.
4. The base RTE is not configurable. This only allocates the base libraries. We do not execute option C (Configure), and proceed to option L (load). The load option populates the RTE libraries defined from the build stage. Review the JCL that is built from entering option L, and issue `SUBMIT`. All condition codes should be 0000.
5. The base RTE is not a complete environment. It needs a configurable RTE that is called the sharing RTE. We supplement our base RTE with the sharing RTE, which will be configured for each of our sysplex environments. We call the sharing RTE for our first SC52 machine `SHR1` as shown in Figure 3-13.

```
------------------------- RUNTIME ENVIRONMENTS (RTEs)-----------------------
COMMAND ===>

Actions: A Add RTE, B Build libraries, C Configure,
         L Load all product libraries after SMP/E,
         D Delete, U Update, V View values, Z Utilities

Action Name     Type    Sharing  Description
  A    SHR1    SHARING BASE    Sharing RTE for SC52
```

Figure 3-13 Sharing RTE creation

6. Figure 3-14 shows the variables for the SHR1 sharing RTE. The Tivoli Enterprise Monitoring Server in this system is called SC52:CMS.

```
----------------------- ADD RUNTIME ENVIRONMENT (1 of 2) -------------------
COMMAND ===>

RTE: SHR1 Type: SHARING Desc: Sharing RTE for SC52

  Libraries  High-level Qualifier       Volser Unit     Storclas Mgmtclas PDSE
  Non-VSAM   OMEGAMON                   OMXE3C 3390                        N
  VSAM       OMEGAMON                   OMXE3C
  Mid-level qualifier ==> SHR1

  JCL suffix          ==> SHR1
  STC prefix          ==> CANS
  SYSOUT class        ==> X            Diagnostic SYSOUT class  ==> X
  Load optimization   ==> N     (Y, N)

  Will this RTE have a Candle Management Server ==> Y  (Y, N)
     If Y, CMS name     ==> SC52:CMS (Case sensitive)

  Copy configuration values from RTE ==>           (Optional)

 Enter=Next  F1=Help  F3=Back
```

Figure 3-14 Sharing RTE properties (1 of 2)

7. Figure 3-15 shows the next page for the sharing RTE properties.

```
----------------------- ADD RUNTIME ENVIRONMENT (2 of 2)--------------------
COMMAND ===>

Use OS/390 system variables?    ==> Y (Y, N)
   RTE name specification       ==> &SYSNAME
   RTE base alias specification ==>
   Applid prefix specification  ==> K&SYSCLONE.
   Use VTAM model applids?      ==> N (Y, N)

Security system         ==> NONE  (RACF, ACF2, TSS, NAM, None)
  ACF2 macro library    ==>

VTAM communications values:
   Applid prefix        ==> CTD         Network ID     ==> USIMBSC
   Logmode table        ==> KDSMTAB1    LU6.2 logmode  ==> CANCTDCS

If you require TCP/IP communications for this RTE, complete these values:
  Hostname              ==> WTSC52
  Address               ==> 9.12.4.42
  Started task          ==> TCPIP
  Port number           ==> 1918
  Interlink subsystem   ==>        (if applicable)
 Enter=Next  F1=Help  F3=Back
```

Figure 3-15 Sharing RTE properties (2 of 2)

> **Note:** We defined our RTE environments to have system variable support, enabling the generation of standard procedures for JCL. By utilizing this support, the components inherit the system variable values for the system on which they are started. These variables are then replaced with actual system-specific values using an extra step for each procedure that creates temporary members in the temporary datasets that are used to override the RKANPAR, RKD2PAR, or RKANCMD datasets. The result is that the software runs using the system-specific parameter values for the particular host z/OS system. For more about system variables, enter `README SYS` on the configuration tool panel.

8. Similar to the base RTE, we must build this SHR1 RTE using the option B (build) and submit the resulting job. This job should complete with condition code 0000.
9. Product component configuration is performed as part of configuring a sharing RTE. When configuring a sharing RTE, the options for configuring

product components are shown. Here, we start configuring the components. Later, after all configuration tasks have been performed, the sharing RTE should be loaded with the option L (load). For the IBM Tivoli OMEGAMON XE on z/OS, the available components are:

– Candle Management Server (configured in 3.3.4, “Configuring Tivoli Enterprise Monitoring Server” on page 62)
– OMEGAMON II for MVS (3.3.5, “Configuring OMEGAMON II for MVS” on page 68)
– IBM Tivoli OMEGAMON XE on z/OS (3.3.6, “Configuring IBM Tivoli OMEGAMON XE on z/OS” on page 73)

The IBM Tivoli OMEGAMON XE on z/OS agent requires a Tivoli Enterprise Monitoring Server address space. This Tivoli Enterprise Monitoring Server will be a remote Tivoli Enterprise Monitoring Server.

> **Note:** Our installation also includes IBM Tivoli OMEGAMON XE for CICS, IMS, Mainframe Networks, Storage, and UNIX System Services. These are installed as separate agents and run in their own address spaces on z/OS.

3.3.4 Configuring Tivoli Enterprise Monitoring Server

Configuring the sharing RTE SHR1 takes us to the Component Selection Menu as shown in Figure 3-16. In this component selection, we recommend that you select the options in the order displayed. We start by configuring Candle Management Server.

```
----------------------- PRODUCT COMPONENT SELECTION MENU--------------------
COMMAND ===> 1

 The following list of components requires configuration to make the product
 operational.  Refer to the appropriate configuration documentation if you
 require additional information to complete the configuration.

 To configure the desired component, enter the selection number on the
command
 line.  You should configure the components in the order they are listed.

      COMPONENT TITLE

  1   Candle Management Server
  2   OMEGAMON II for MVS
  3    IBM Tivoli OMEGAMON on z/OS
```

Figure 3-16 Component selection menu

As shown in Figure 3-17, this configuration option performs the following tasks:

- Create the logical unit 6.2 logmode
- Specify configuration values
- Specify communication protocols
- Create runtime members
- Configure the historical datastore

```
------------- CONFIGURE THE CMS (V360) / RTE: SHR1 ------------- -----------
OPTION ===> 1

 Each RTE can contain only one CMS.  To configure          Last selected
 the CMS for this RTE, perform these steps in order:        Date     Time

   1  Create LU6.2 logmode                                 05/11/01  16:56
   2  Specify configuration values                         05/11/01  17:07
   3  Specify communication protocols                      05/11/01  17:17
   4  Create runtime members                               05/11/02  10:33
   5  Configure persistent datastore                       05/11/01  17:28
   6  Complete the configuration                           05/11/01  17:31

 Optional:

   7  View CMS list and registration status                 05/11/01  17:32
```

Figure 3-17 Configuration option list

1. We follow the sequence of the configuration options in Figure 3-17, then configure the LU6.2 log mode as shown in Figure 3-18. This option creates a job, which should be submitted and completed with condition code 0000.

```
----------------------------- CREATE LU6.2 LOGMODE--------------------------
COMMAND ===>

 The CMS requires an LU6.2 logmode.  Complete the items on this panel and
 press Enter to create a job that will assemble and link the required logmode.

   LU6.2 logmode        ==> CANCTDCS
   Logmode table name   ==> KDSMTAB1

   VTAMLIB load library ==> SYS1.LOCAL.VTAMLIB
   VTAM macro library   ==> SYS1.SISTMAC1
Enter=Next  F1=Help  F3=Back
```

Figure 3-18 LU 6.2 logmode configuration

2. For the configuration values with option 2 as shown in Figure 3-19. We define SC52 as a hub Tivoli Enterprise Monitoring Server.

```
------------------------ SPECIFY CONFIGURATION VALUES ---------------------
COMMAND ===>

 To configure a CMS for your site, complete the items on this panel.

   CMS started task      ==> CANSDSST
   Hub or Remote?        ==> HUB
   Security validation?  ==> N        (Y, N)
Enter=Next  F1=Help  F3=Back  F5=Advanced
```

Figure 3-19 Defining a hub Tivoli Enterprise Monitoring Server

3. Using the advanced option with F5, we changed the minimum extended storage from the default value 150000K to 250000K as shown in Figure 3-20. This value should be adjusted depending on your environment. This ensures that the Tivoli Enterprise Monitoring Server has enough storage allocated to its storage pools.

```
-------------------- SPECIFY ADVANCED CONFIGURATION VALUES -----------------
COMMAND ===>

 Advanced CMS configuration options:
   Enable startup console messages ==> Y      (Y, N)
   Enable communications trace     ==> N      (Y, N)
   Enable storage detail logging   ==> Y      (Y, N)
   Intervals (hh:mm):
     Storage detail logging: Hours ==> 0  (0-24)  Minutes ==> 60 (0-60)
     Flush VSAM buffers:     Hours ==> 0  (0-24)  Minutes ==> 30 (0-60)
   Minimum extended storage        ==> 250000  K
   Maximum storage request size    ==> 16     (Primary)
   Maximum storage request size    ==> 23     (Extended)

 Persistent datastore parameters:
   Maintenance procedure prefix     ==> KPDPROC
   Datastore file high-level prefix ==> OMEGAMON.SHR1
   Volume  ==> OMXE3C     Storclas  ==>
   Unit    ==> 3390       Mgmtclas  ==>
```

Figure 3-20 Advanced configuration for Tivoli Enterprise Monitoring Server

4. For the communication protocol in option 3, we configure communication as shown in Figure 3-21. You should have one protocol to be defined as SNA.

```
----------------------- SPECIFY COMMUNICATION PROTOCOLS---------------------
COMMAND ===>

 Specify communication protocols in priority sequence for
 CMS SC52:CMS.

     Protocol 1                ==> IPPIPE    (SNA, IP, IPPIPE)
     Protocol 2 (optional)     ==> IP        (SNA, IP, IPPIPE)
     Protocol 3 (optional)     ==> SNA       (SNA, IP, IPPIPE)
```

Figure 3-21 Communication protocol

– Figure 3-22 shows the IP.PIPE or TCP communication option.

```
-------------------- SPECIFY IP.PIPE COMMUNICATION PROTOCOL-----------------
COMMAND ===>

 Specify the TCP communication values for this CMS.

   Hostname               ==> WTSC52
   Address                ==> 9.12.4.42
   Started task           ==> TCPIP

   If applicable
     Network interface card (NIC) ==>
     Interlink subsystem          ==>
     IUCV interface in use?       ==> N     (Y, N)

 Specify IP.PIPE configuration
   Port number            ==> 1918
   Address translation    ==> N     (Y, N)
      Partition name      ==>
```

Figure 3-22 IP PIPE communication option

– Figure 3-23 shows the IP or UDP communication option.

```
--------------------- SPECIFY IP COMMUNICATION PROTOCOL -------------------
COMMAND ===>

 Specify the TCP communication values for this CMS.

   Hostname               ==> WTSC52
   Address                ==> 9.12.4.42
   Started task           ==> TCPIP

   If applicable
     Network interface card (NIC) ==>
     Interlink subsystem          ==>
     IUCV interface in use?       ==> N     (Y, N)

 Specify IP configuration
   Port number             ==> 1918
```

Figure 3-23 IP communication option

– Figure 3-24 shows the SNA communication option. We also need to update the application IDs using F6.

```
--------------------- SPECIFY SNA COMMUNICATION PROTOCOL ------------------
COMMAND ===>

 Specify the SNA communication values for this CMS.

   Applid prefix        ==> K&SYSCLONE.DS
   Network ID           ==> USIBMSC (NETID value from SYS1.VTAMLST(ATCSTRnn)

Enter=Next  F1=Help  F3=Back  F6=Applids
```

Figure 3-24 SNA communication option

– Figure 3-25 shows the VTAM APPLID values.

```
------------------------- SPECIFY VTAM APPLID VALUES ----------------------
COMMAND ===>

   Specify the VTAM node and applid information on this panel:

     Major node                       ==> K&SYSCLONE.DSN
     Local location broker            ==> K&SYSCLONE.DSLB
     CMS application ID               ==> K&SYSCLONE.DSDS
     CT/Engine operator               ==> K&SYSCLONE.DSOP
     CT/Engine operator (non-CUA)     ==> K&SYSCLONE.DSOR
     CT/Engine VTAM program operator  ==> K&SYSCLONE.DSVP
     Outbound seeding                 ==> K&SYSCLONE.DSOB
     PDS primary maintenance          ==> K&SYSCLONE.DSP1
     PDS secondary maintenance        ==> K&SYSCLONE.DSP2
     MQ/3270 report                   ==> K&SYSCLONE.DSMQ
     MQ/3270 Japanese report          ==> K&SYSCLONE.DSMJ
     Alert Adapter                    ==> K&SYSCLONE.DSAA
```

Figure 3-25 VTAM applid list

5. Defining the runtime member from option 4 of the Tivoli Enterprise Monitoring Server configuration generates a job that you should submit. The job should have a condition code of 0000.
6. Option 5 to configure the persistent data store allows the configuration of historical data. The persistent data store configuration has several sub-steps as shown in Figure 3-26. We follow each option there.

```
---------------------- ALLOCATE PERSISTENT DATASTORE MENU ------------------
OPTION ===>
                                                           Last selected
 Perform these configuration steps in order:                Date     Time

   1  Modify and review datastore specifications           05/11/01  17:25
   2  Create or edit PDS maintenance jobcard               05/11/01  17:26
   3  Create runtime members                               05/11/01  17:28
   4  Edit and submit datastore allocation job             05/11/01  17:29
   5  Complete persistent datastore configuration          05/11/01  17:29
```

Figure 3-26 Persistent data store configuration

7. The last option from Tivoli Enterprise Monitoring Server configuration explains the tasks to complete the configuration outside of the configuration tool. This step requires you to perform several tasks. These are the tasks that we performed in our setup:

   a. Copy the Candle Management Server started task CANSDSST from RKANSAM to PROCLIB.

   b. Copy persistent data store procedures to PROCLIB:

      - KPDPROC1: Sample PDS maintenance procedure
      - KPDPROC2: Batch job procedure to back up, export,

   c. Create system variable members. We enabled system variable support, so we ran the CB#VSHR1 system variable members job to create the system variable parameter member and other components.

   d. Copy VTAM definitions to VTAMLST and vary VTAM major node active. Use this console command:

      ```
      V NET,ACT,ID=K52DSN
      ```

   e. APF-authorize libraries are libraries from our base RTE: OMEGAMON.BASE.RKANMOD and OMEGAMON.BASE.RKANMODL

      Note: You can dynamically add the above data sets to your APF list by using these commands:

      ```
      SETPROG APF,ADD,DSN=OMEGAMON.BASE.RKANMOD,VOL=OMXE3C
      SETPROG APF,ADD,DSN=OMEGAMON.BASE.RKANMODL,VOL=OMXE3C
      ```

   f. Verify installation and configuration. Start the Candle Management Server started task using this command:

      ```
      S CANSDSST
      ```

      The output from the CANSDSST task should contain these messages:

      ```
      KDSMA001   Candle Management Server (CMS) data collection
      server started.
      KO4SRV032  Candle Management Server (CMS) startup complete.
      ```

3.3.5 Configuring OMEGAMON II for MVS

Next, set up the OMEGAMON II for MVS. The configuration of OMEGAMON II for MVS requires several steps. You must specify configuration values, create runtime members, and register the OMEGAMON II for MVS parameters with the Tivoli Enterprise Monitoring Server configured in this RTE. This also upgrades any previous version of OMEGAMON II for MVS in this RTE automatically.

For a new RTE, you must also install the Candle subsystem, a z/OS subsystem that runs in its own address space and enables OMEGAMON IIs in other address spaces to monitor dynamic I/O information. A single Candle subsystem can support multiple OMEGAMON II address spaces. The Candle subsystem requires the KCNDLINT module to be in the link list.

Note: Be sure you have the latest version of the Candle subsystem running; otherwise, unpredictable results may occur.

Figure 3-27 shows the configuration of OMEGAMON II for MVS.

```
---------------- CONFIGURE OMEGAMON II FOR MVS / RTE: SHR1 -----------------
OPTION ===> 1
                                                            Last selected
 Perform these configuration steps in order:                 Date     Time

   1  Specify configuration values                          05/11/14  15:40
   2  Allocate historical datastores                        05/11/14  15:40
   3  Create runtime members                                05/11/02  10:34
   4  Modify Classic interface command security             05/11/14  15:39
   5  Register with local CMS                               05/11/14  15:39
   6  Install Candle Subsystem                              05/11/03  10:17
   7  Complete the configuration

 Optional:

   8  Run migration utility
```

Figure 3-27 OMEGAMON II for MVS configuration list

1. Specifying the configuration values with option 1 from Figure 3-27 on page 69, we get the dialog shown in Figure 3-28.

```
------------------------- SPECIFY CONFIGURATION VALUES ---------------------
COMMAND ===>

   Started tasks:

     OMEGAMON II for MVS                        ==> CANSM2
     Realtime collector OMEGAMON                ==> CANSM2RC
     CSA analyzer                               ==> CANSM2CS
     Historical collector (EPILOG)              ==> CANSM2HI
     Historical data interface                  ==> CANSM2HD
     Zoom-to-EPILOG                             ==> CANSM2EZ
     End-to-End (ETE)                           ==> CANSETE

   VTAM and OMEGAMON II information:

     Applid prefix           ==> K&SYSCLONE.M2
     Virtual terminal prefix  ==> K&SYSCLONE.M2
     Maximum number of CUA users                ==> 99      (10-256)
     Maximum number of collector users (UMAX) ==> 99      (1-99)
     Enable WTO messages                        ==> N       (Y, N)

 Enter=Next  F1=Help  F3=Back  F6=Applids
```

Figure 3-28 OMEGAMON II for MVS definition

2. Figure 3-29 shows what we specified for the Historical Datastore parameters.

```
V550 --------------- SPECIFY HISTORICAL DATASTORE ALLOCATION ---------------
COMMAND ===>

 Specify dataset names for new historical VSAM clusters, then press Enter.
 If you are migrating from a prior version and plan to use existing
 datastores, leave this panel blank.
 Note: Make sure you include the mid-level qualifiers for this RTE.

   Cluster Name (Fully-qualified, no quotes)  Volser  Storclas  Mgmtclas
   OMEGAMON.SHR1.RKM2EDS1________________     OMXE3C  ________  ________
   OMEGAMON.SHR1.RKM2EDS2________________     OMXE3C  ________  ________
   OMEGAMON.SHR1.RKM2EDS3________________     OMXE3C  ________  ________
   ______________________________________     ______  ________  ________
   ______________________________________     ______  ________  ________

   Primary space   ==> 50  cylinders          Secondary space ==> 0   cyls

   Optional 10-cylinder Profile Datastore     Volser  Storclas  Mgmtclas
   ______________________________________     ______  ________  ________
```

Figure 3-29 Historical data store parameters

3. Figure 3-30 shows the values we specified for security. (We selected `NONE`.)

```
-------------------- SPECIFY SECURITY AND OTHER PARAMETERS -----------------
COMMAND ===>

 Complete the items on this panel with values for your site.

 OMEGAMON II logon security   ==> NONE     (RACF, ACF2, TSS, NAM, None)

 Historical datastore maintenance:
   Primary proc name          ==> CANSM2HP
   Secondary proc name        ==> CANSM2HS
   Operating mode             ==> RESTART  (BOTH, RESTART, STOP, None)
   Backup tape unit           ==>
   Backup retention in days   ==>

 Historical batch report job  ==> CANSM2BA
```

Figure 3-30 Candle security

4. Next we submit the jobs in option 2 to 5 from the menu in Figure 3-27 on page 69. Each step should be completed successfully before moving to subsequent step.

5. In option 6 to Install Candle Subsystem, we specify the configuration parameter shown in Figure 3-31.

```
---------------------- SPECIFY CONFIGURATION PARAMETERS ----------------------
COMMAND ===>

 Specify the Candle Subsystem parameters.
 Note:  The default subsystem ID is CNDL.  If you change it, be sure to
        read the information under Complete the configuration.

    Subsystem ID    ==> CNDL

    Started task    ==> CANSCN

    IEFSSN00 format ==> K  (K=keyword, P=positional)


 Specify the Coupling Facility collection parameters.
 Note:  Every Candle Subsystem configured within a Sysplex must use the
        same XCF group name value.

    XCF group name       ==> KCNXCFCF

    IXCQUERY WTO message ==> ERROR  (No, All, Error)

 Enter=Next  F1=Help  F3=Back
```

Figure 3-31 Candle subsystem

6. After we submit the job to define the subsystem members, we are required to complete the configuration outside of the configuration tool:
   – Copy the Candle Subsystem started task, CANSCN, from RKANSAM to PROCLIB. You may change this name to any JCL procedure name that meets your installation standards. However, do not use the name of the Candle Subsystem (subsystem ID) as the name of your JCL procedure.
   – Update the IEFSSNxx member of SYS1.PARMLIB to define the Candle Subsystem. The subsystem entry should be similar to Example 3-2.

Example 3-2 Candle subsystem

```
SUBSYS SUBNAME(CNDL)
  INITRTN(KCNDLINT)
```

> **Note:** You can dynamically add the subsystem name using this command:
>
> ```
> SETSSI ADD,SUB=CNDL,I=KCNDLINT
> ```
>
> You can start the CANSCN started task automatically by adding the INITPARM definition of SSPROC=CANSCN. However, this might start the CANSCN too early in the IPL process. This parameter is case sensitive.

7. During the complete the configuration step (option 7 on Figure 3-27 on page 69), we copy the required STC procedures from RKANSAM to SYS1.PROCLIB for OMEGAMON II for MVS:

CANSM2	OMEGAMON II for MVS
CANSM2RC	Realtime collector OMEGAMON
CANSM2CS	CSA analyzer
CANSM2HI	Historical collector (EPILOG)
CANSM2HD	Historical data interface
CANSM2EZ	Zoom-to-EPILOG
CANSM2HP	Primary proc name
CANSM2HS	Secondary proc name
CANSETE	End-to-End response time watch

3.3.6 Configuring IBM Tivoli OMEGAMON XE on z/OS

After completing the setup for OMEGAMON II, we selected option 3 to configure the OMEGAMON for z/OS component. In this step you define and configure sysplex-level entities, assign the RTE to a sysplex, install product-specific information on the Tivoli Enterprise Monitoring Server, and register the IBM Tivoli OMEGAMON XE on z/OS agent in the Tivoli Enterprise Monitoring Server address space.

You also specify which RTEs and, by extension, which Tivoli Enterprise Monitoring Servers will be eligible to serve as the sysplex proxy, the data consolidation point for sysplex information gathered by the monitoring agents.

If you have at least one IBM crytographic coprocessor installed and want to monitor its status and performance, configure and install the ICSF exit that IBM Tivoli OMEGAMON XE on z/OS uses to collect cryptographic coprocessor data.

You can also configure the historical datastore for IBM Tivoli OMEGAMON XE on z/OS historical data collection. This step configures the persistent datastores and the product-specific datasets that are used to store sysplex and system level data. This is necessary if you intend to enable historical data collection and reporting (using the Tivoli Enterprise Portal) and must be completed before you can enable data warehousing.

If you are running one of the products superseded by IBM Tivoli OMEGAMON XE on z/OS (OMEGAMON for OS/390 Version 140, OMEGAMON for Sysplex Version 220, OMEGAMON for IBM Cryptographic Coprocessors Version 100) in the RTE, it will be upgraded automatically when you configure the agent. Any custom situations will be migrated when you seed the hub Tivoli Enterprise Monitoring Server. Any custom workspaces will be migrated when you seed the Tivoli Enterprise Portal.

Our sharing RTE SHR1 is part of a sysplex-wide RTE, so we start with option 1 as shown in Figure 3-32.

```
---------------- CONFIGURE OMEGAMON on z/OS / RTE: SHR1 -----------------
OPTION ===>  1
                                                           Last selected
 Perform these configuration steps in order:                Date     Time

   0  Define a single LPAR environment         (Optional)
      Note: To configure a single LPAR, non-Sysplex
      environment, select option 0 then skip options
      1 and 2.  To configure a Sysplex, skip option 0.

   1  Define Enterprise wide ENQ manager       (Optional) 05/11/10  11:30
   2  Define your Sysplex environment                     05/11/10  10:14

   3  Specify configuration parameters for ICSF (Optional) 05/11/01  17:57

   4  Assign this RTE to a Sysplex                        05/11/01  18:02

   5  Register with local CMS                             05/11/08  18:53
   6  Install Agent into local CMS                        05/11/01  18:06
   7  Configure persistent datastore (in CMS)             05/11/10  10:55

   8  Complete the configuration                          05/11/01  18:12
 F1=Help  F3=Back
```

Figure 3-32 OMEGAMON configuration menu

With option 1, we can define our enqueue in sysplex (ENQplex). An ENQplex is a group of z/OS images in a sysplex environment under common enqueue management. If you do not define any ENQplexes, the configuration tool creates a default ENQplex called $DEFAULT. You can use this option to define ENQplexes later. Changes here are global for all sysplex and RTE definitions. We defined our ENQplex at this point, and called it `ITSOENQ` (Figure 3-33). The RTE count is based on the final number of RTEs defined.

```
--------------------- Define Enterprise wide ENQ manager ------------------
COMMAND ===>

 Action: A Add a new ENQPlex environment
         D Delete an ENQPlex environment
         V View list of Sysplexs assigned to an ENQPlex

Action ENQPlex                                          Sysplex   RTE
code   name     Description                              count    count
 _     ________ ______________________________________
 _     $DEFAULT DEFAULT ENQPLEX ENVIRONMENT                0        0
 _     ITSOENQ  ITSO Enqueue Plex                          1        3
```

Figure 3-33 ENQplex definition

We complete steps 2 - 7 (Figure 3-32 on page 74) by submitting the generated jobs. These jobs should complete successfully.

3.3.7 Loading the RTE and completing configuration

The last step on the sharing RTE definition is to load it with the configured members (Figure 3-34).

```
------------------------ RUNTIME ENVIRONMENTS (RTEs) ----------------------
COMMAND ===>

Actions: A Add RTE, B Build libraries, C Configure,
         L Load all product libraries after SMP/E,
         D Delete, U Update, V View values, Z Utilities

Action Name     Type     Sharing  Description
  _    ________ ________ ________ _____________________________________________
  _    BASE     BASE              Base RTE for SC52, SC69, SC67
  1    SHR1     SHARING BASE      Sharing RTE for SC52
```

Figure 3-34 Loading the sharing RTE

After the LOAD job has completed successfully, we started Tivoli Enterprise Monitoring Server and OMEGAMON II address spaces. Figure 3-35 shows the started tasks in SDSF.

```
 Display  Filter  View  Print  Options  Help
-------------------------------------------------------------------------------
SDSF DA SC52  SC52     PAG    0 SIO   399 CPU  10/ 10  LINE 1-9 (9)
COMMAND INPUT ===>                                              SCROLL ===> PAGE
NP   JOBNAME  StepName ProcStep JobID    Owner    C Pos DP Real Paging
     CANSM2RC CANSM2RC M2COLL   STC16914 STC        NS  FE 5161   0.00
     CANSCN   CANSCN   CNDL     STC16902 STC        NS  FE  337   0.00
     CANSM2CS CANSM2CS CSAA     STC16915 STC        NS  FE  318   0.00
     CANSDSST CANSDSST CMS      STC19883 STC        NS  FE  31T   0.00
     CANSM2EZ CANSM2EZ EPZOOM   STC16920 STC        NS  FE  405   0.00
     CANSM2HD CANSM2HD HDI      STC16918 STC        LO  FF  363   0.00
     CANSETE  CANSETE  ETESTART STC16913 STC        NS  FE  248   0.00
     CANSM2   CANSM2   CANSM2   STC16912 STC        NS  FE 5739   0.00
     CANSM2HI CANSM2HI EPC      STC16919 STC        NS  FE 4115   0.00
```

Figure 3-35 IBM Tivoli OMEGAMON XE on z/OS started tasks

Note: Update the CANSM2 procedure to automatically start the other started tasks by uncommenting the start commands at the beginning of the procedure:

```
// VARY NET,ACT,ID=K&SYSCLONE.M2N,SCOPE=ALL
// START CANSETE
// START CANSM2CS
// START CANSM2RC
// START CANSM2HD
// START CANSM2HI
// START CANSM2EZ
```

3.4 Tivoli Enterprise Portal Server implementation

This section discusses the implementation of the Tivoli Enterprise Portal Server in the context of IBM Tivoli OMEGAMON XE V3.1 installation. The Tivoli Enterprise Portal Server used in this section is version 1.9.6.2 (V1.9.6 with fixpack 2). We installed this server on a Windows-based system. We also used this Tivoli Enterprise Portal Server to seed the hub Tivoli Enterprise Monitoring Server. The discussion consists of:

- 3.4.1, “Required software” on page 77
- 3.4.2, “Windows systems IBM Tivoli Monitoring Services” on page 78
- 3.4.3, “Install support for IBM Tivoli OMEGAMON XE” on page 84

- 3.4.4, “Seeding our z/OS hub Tivoli Enterprise Monitoring Server” on page 85
- 3.4.5, “Logging on to our Tivoli Enterprise Portal” on page 89

3.4.1 Required software

This software is required for installing Tivoli Enterprise Portal Server V196:

- Sun Microsystem Java Runtime Environment V1.3.1_04 through V1.3.1_14 or V1.4.2_01 through V1.4.2_08.
- DB2 UDB V8.1 or V8.2; or Microsoft® SQL Server V7.0 or 2000

> **Note:** See *Installing and Setting up OMEGAMON Platform and CandleNet Portal on Windows and UNIX*, SC32-1768, at:
>
> http://publib.boulder.ibm.com/tividd/td/CanInstallDoc/SC32-1768-01/en_US/PDF/SC32-1768-01.pdf

Figure 3-36 shows our setup of where we installed our Tivoli Enterprise Portal Server and Tivoli Enterprise Monitoring Server, along with components installed on the system. System SC52 contains our hub Tivoli Enterprise Monitoring Server, with remote Tivoli Enterprise Monitoring Server on SC67 and SC69.

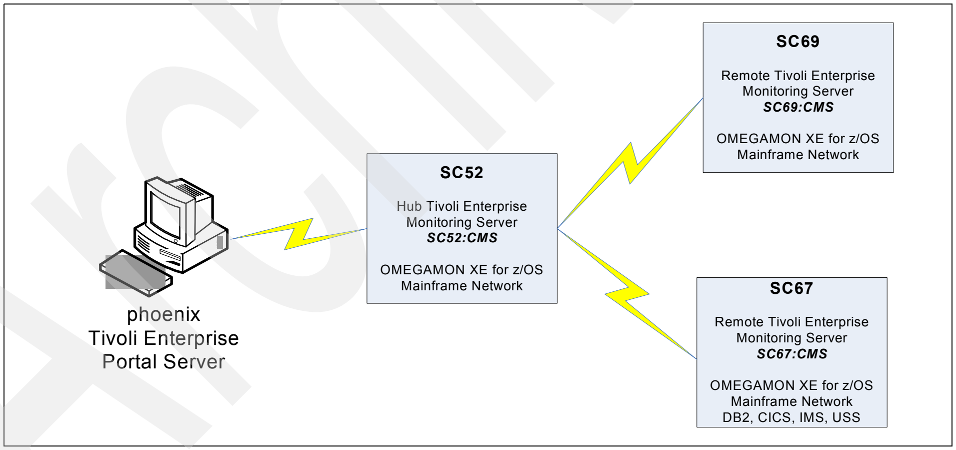


Figure 3-36 Our Tivoli Enterprise Portal Server and Tivoli Enterprise Management Server configuration

3.4.2 Windows systems IBM Tivoli Monitoring Services

The installation procedure for Tivoli Enterprise Portal Server and Tivoli Enterprise Portal clients on the Windows platform follows these steps:

1. Invoke Tivoli Enterprise Portal Server installation using **`setup.exe`** from the IBM Tivoli OMEGAMON Platform CD v3.6.1 (LCD7-0707-01). The CD-ROM, or platform CD, contains Tivoli Enterprise Monitoring Server V360, Tivoli Enterprise Portal Server V196, Tivoli Enterprise Portal, and additional agent supports.
2. Figure 3-37 shows that we used the default installation directory for Tivoli Enterprise Portal Server, which is C:\Candle\.

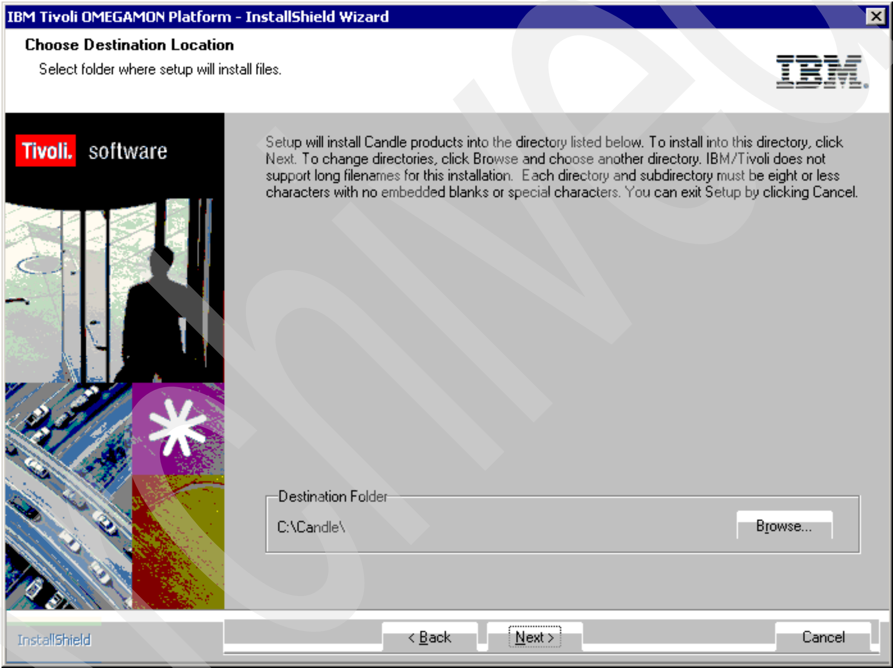


Figure 3-37 Installation directory

3. From the product features shown in Figure 3-38, select the following products:
   - CandleNet Portal Server
   - CandleNet Portal Desktop Client
   - Candle Management Workstation

   We did not select the agent support file, as we want to use the most current agent support file provided with IBM Tivoli OMEGAMON XE V3.1.0. We implement this in 3.4.3, “Install support for IBM Tivoli OMEGAMON XE” on page 84.

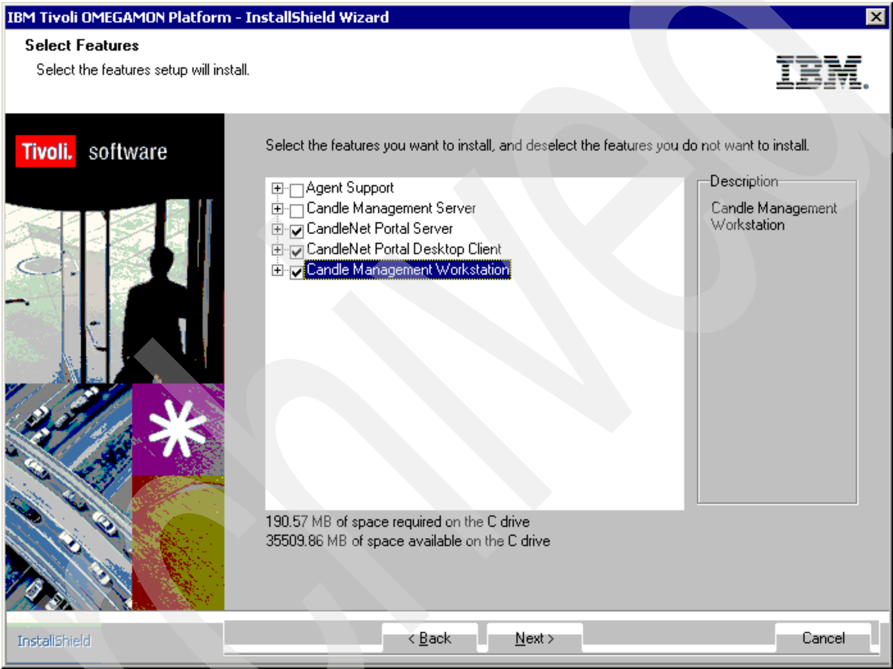


Figure 3-38 Product component selection

4. Figure 3-39 shows the installation summary. Click **Next** to start the installer copying files.

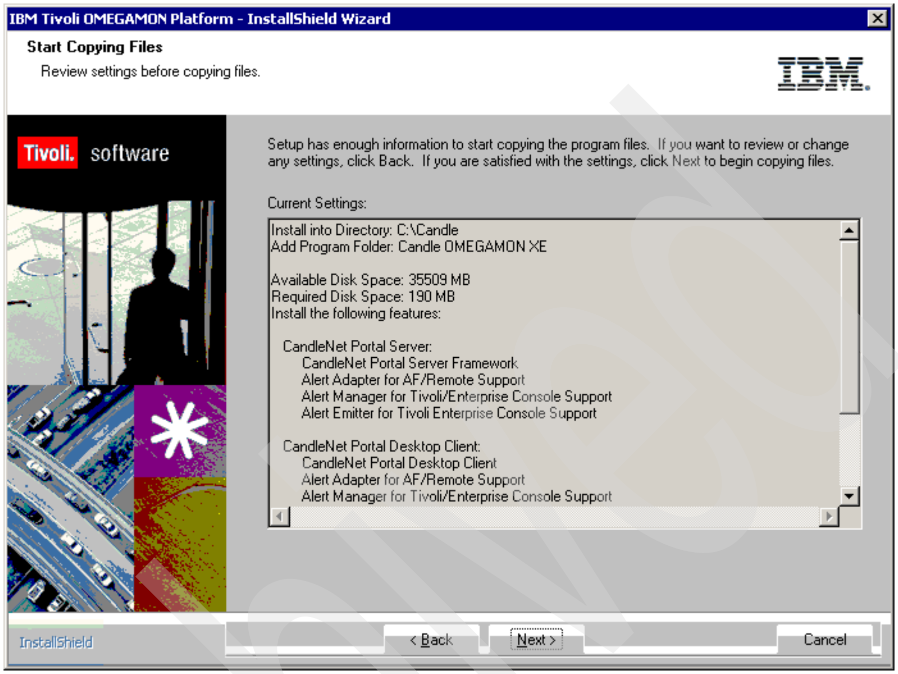


Figure 3-39 Installation summary

5. When the installer has completed the process, you can start configuring the product as shown in the option in Figure 3-40.

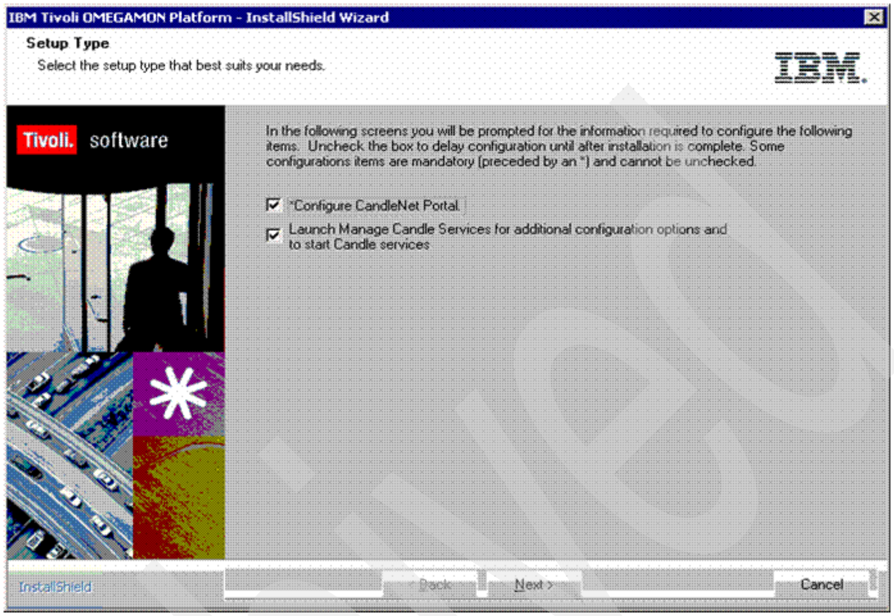


Figure 3-40 Configuration option

6. Setting up for Tivoli Enterprise Portal Server relates to connecting and configuring the DB2 database that is the repository for Tivoli Enterprise Portal Server. Figure 3-41 shows the prompt. Supply the DB2 user ID and password, and the table owners for the Tivoli Enterprise Portal Server tables. (The default is CNPS.) Click **OK**.

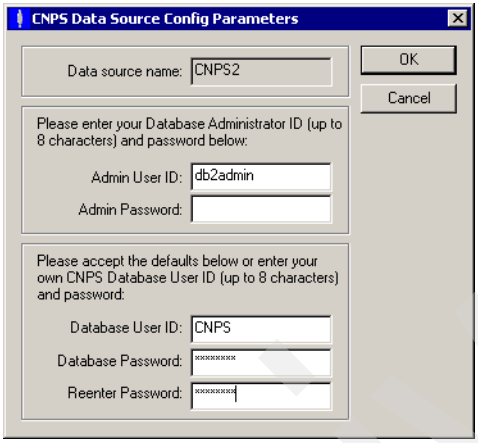


Figure 3-41 DB2 parameters prompt

7. Define the connection to Tivoli Enterprise Monitoring Server from our Tivoli Enterprise Portal Server as shown in Figure 3-42. Click **OK**.

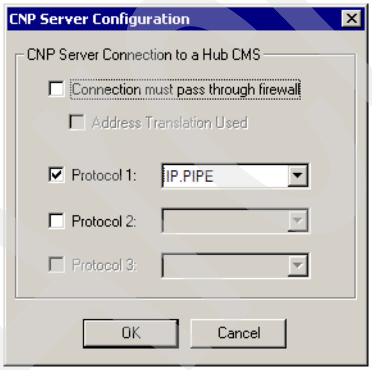


Figure 3-42 Communication options

8. In the next CNP server configuration window (Figure 3-43), specify the connection detail. The Tivoli Enterprise Monitoring Server host name is WTSC52.ITSC.IBM.COM, our z/OS hub Tivoli Enterprise Monitoring Server.

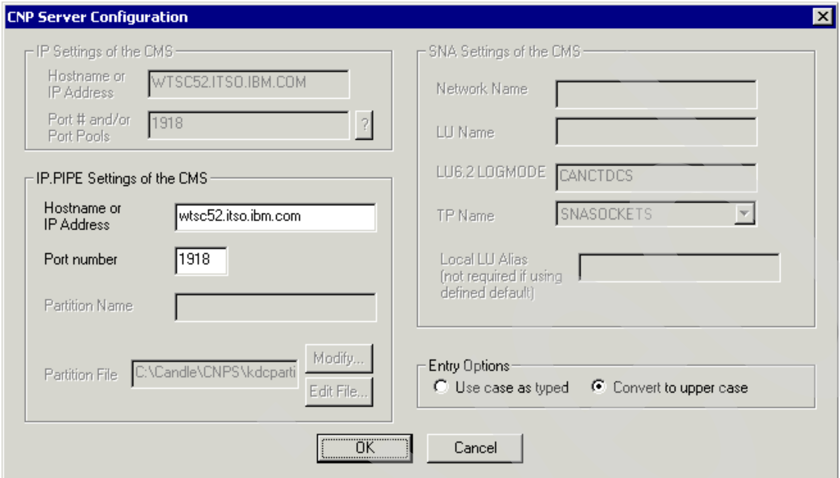


Figure 3-43 Communication option detail

This completes the Tivoli Enterprise Portal Server installation. We were presented with the Manage Candle Services window shown in Figure 3-44.

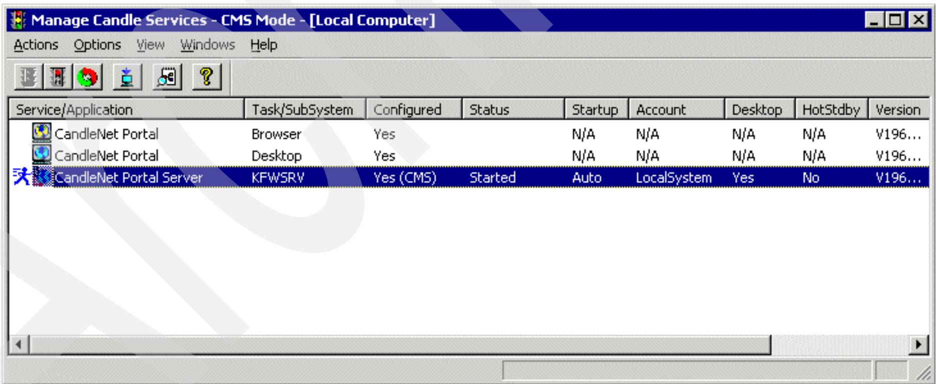


Figure 3-44 Manage Candle services

3.4.3 Install support for IBM Tivoli OMEGAMON XE

After you install and configure a Tivoli Enterprise Portal Server, you should install support for IBM Tivoli OMEGAMON XE on the Tivoli Enterprise Portal Server. The following tasks show our installation for the IBM Tivoli OMEGAMON XE application support files:

1. Insert the IBM Tivoli OMEGAMON XE Data Files for z/OS CD (LCD7-0764-01) into your CD drive. Installation should begin automatically; otherwise, run **`setup.exe`** from the CD.
2. The InstallShield wizard starts; follow it. Figure 3-45 shows the product selection window, which lists the OMEGAMON application support components on the CD-ROM. Expand the + sign next to CandleNet Portal Server, CandleNet Portal Browser Client, and CandleNet Portal Desktop Client components, and check z/OS support for each.

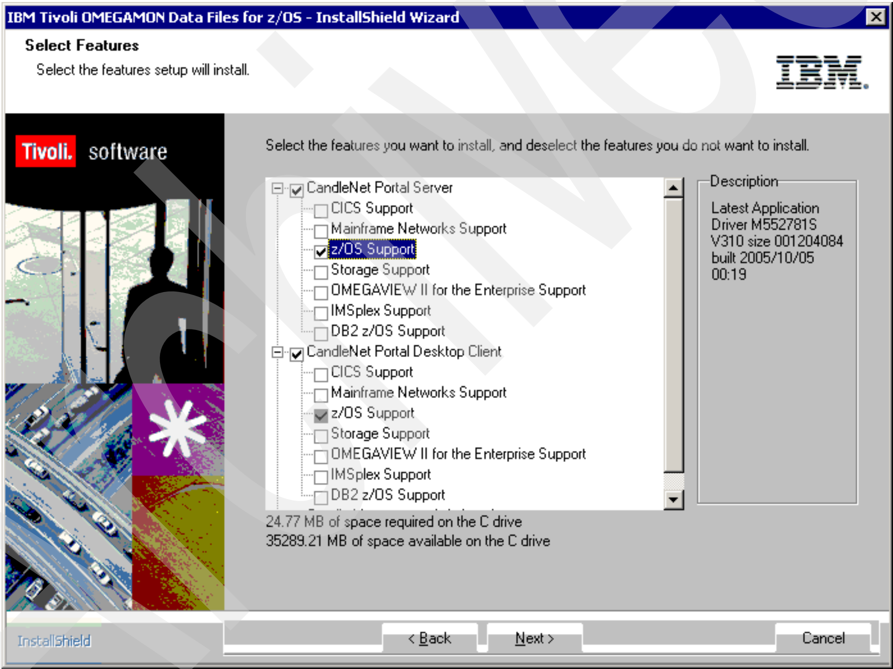


Figure 3-45 Application support

3. Verify the selection on the installation summary window shown in Figure 3-46.

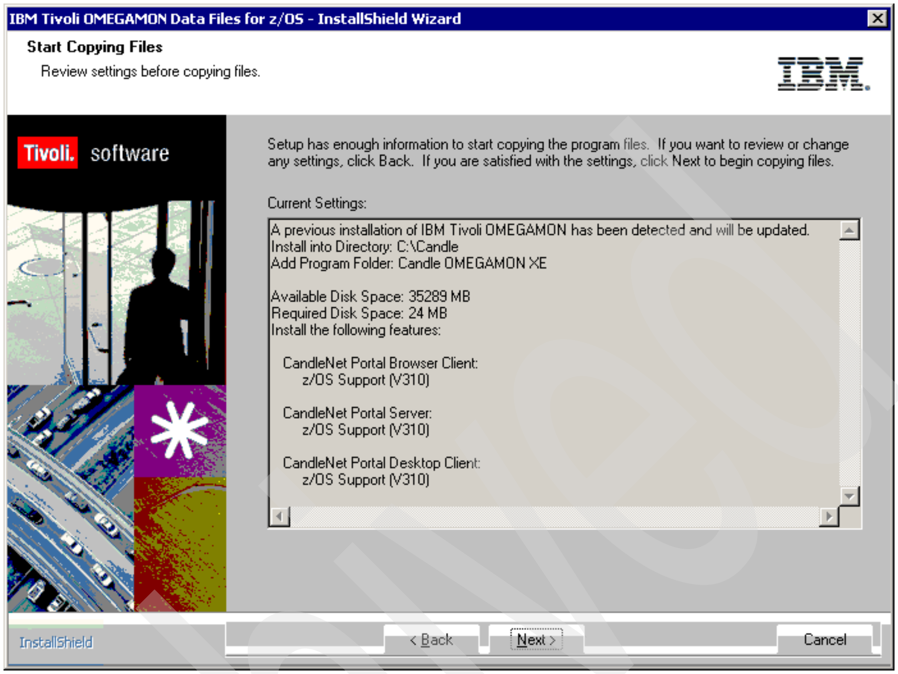


Figure 3-46 Installation summary window

The IBM Tivoli OMEGAMON XE application support files are installed. This installation includes the seed data for the selected components.

3.4.4 Seeding our z/OS hub Tivoli Enterprise Monitoring Server

Before we can use an IBM Tivoli OMEGAMON XE portal monitoring agent, the Tivoli Enterprise Monitoring Server hub must be seeded with the application support files. Remote Tivoli Enterprise Monitoring Servers are seeded when an agent is registered with the local server through the configuration tool or when they connect their agent to the hub Tivoli Enterprise Monitoring Server. Hub servers must be seeded manually from a distributed workstation using the data CD included in the product package. Seeding adds product-provided situations, templates, and other sample data to the hub's Enterprise Information Base (EIB) tables. Hubs on z/OS are seeded from the Tivoli Enterprise Portal Server.

Tivoli Enterprise Monitoring Server must be running during seeding and you will need to stop and restart the server when you have finished seeding it. To seed our hub Tivoli Enterprise Monitoring Server on z/OS we performed the following tasks:

1. Ensure that the hub Tivoli Enterprise Monitoring Server is running. Our hub Tivoli Enterprise Monitoring Server is the started task called CANSDSST.
2. From the machine on which Tivoli Enterprise Portal Server is installed, run Manage Candle Services. For Windows, use **Programs** → **Candle OMEGAMON XE** → **Manage Candle Services**.
3. From the Actions menu in the Manage Candle Services window, select **Actions** → **Advanced** → **Seed CMS** as shown in Figure 3-47.

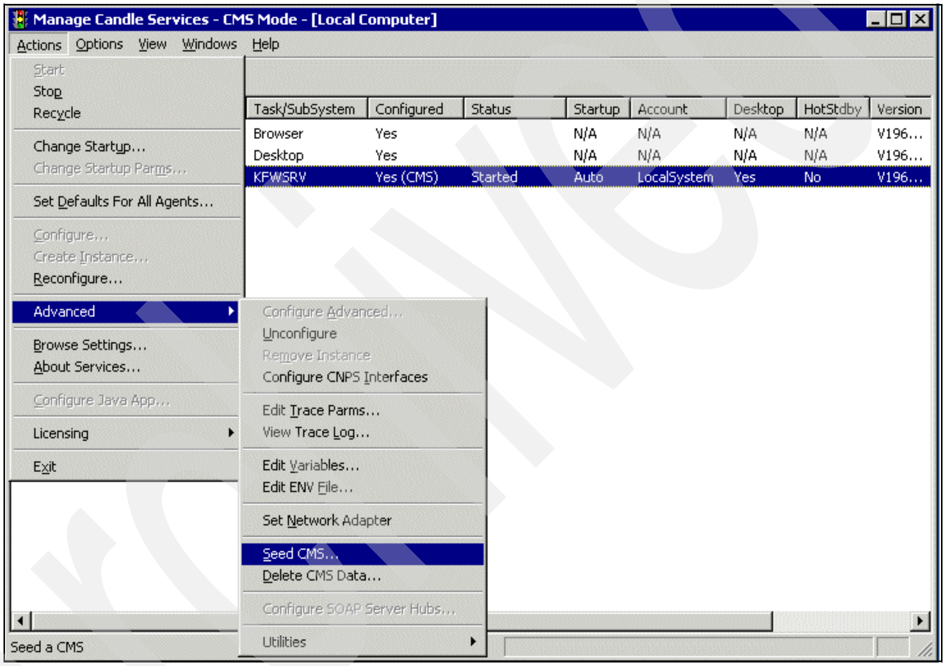


Figure 3-47 Seeding z/OS Tivoli Enterprise Monitoring Server

4. On the Seed CMS window, select **On a different computer** and click **OK** (see Figure 3-48). Click **OK** again when you are prompted to confirm that the Candle Management Server is configured and running.

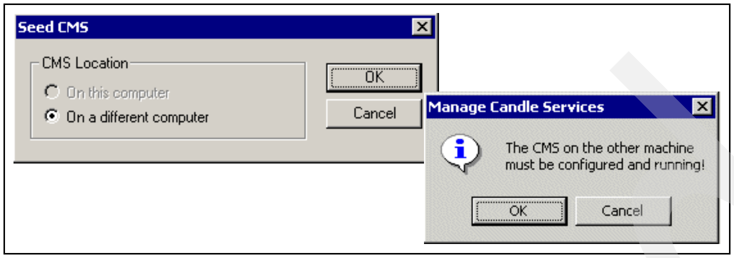


Figure 3-48 Seeding the Tivoli Enterprise Monitoring Server

5. On the Non-Resident CMS Connection window, provide the Tivoli Enterprise Monitoring Server node ID as shown in Figure 3-49. Here we enter our Tivoli Enterprise Monitoring Server node of `SC52:CMS`. Select the appropriate communications protocol and click **OK**.

> **Note:** You can find the Node ID as the value of the CMS_NODEID variable in the dataset: &shilev.&rtename.RKANPAR(KDSENV)

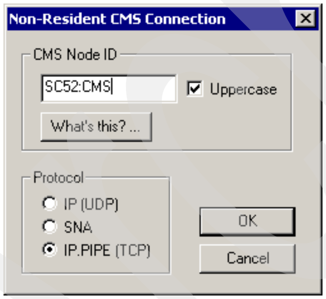


Figure 3-49 Tivoli Enterprise Monitoring Server connection option

6. On the next window (Figure 3-50), we provide the values required by your communications protocol. Our protocol is TCP/IP, so we enter the TCP/IP host name and port number of the Candle Management Server to be seeded (on z/OS).

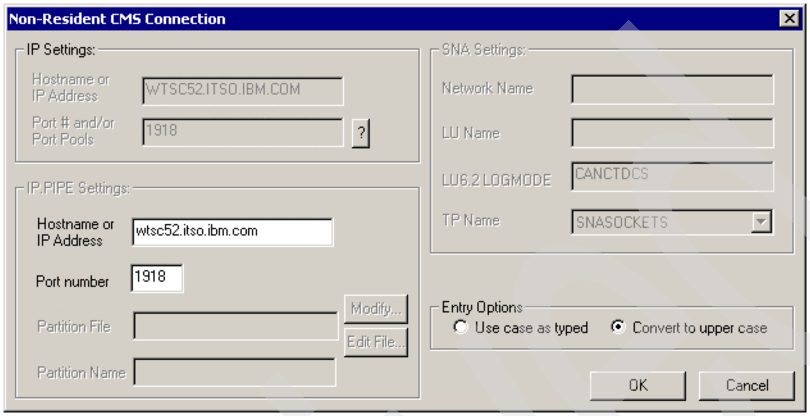


Figure 3-50   Connection to Tivoli Enterprise Monitoring Server detail

7. On the Select Product to Seed CMS window, select the products whose seed data you want to add to the Tivoli Enterprise Monitoring Server configuration, and click **OK**. In our install we select Tivoli Enterprise Monitoring Server data for z/OS support as shown in Figure 3-51.

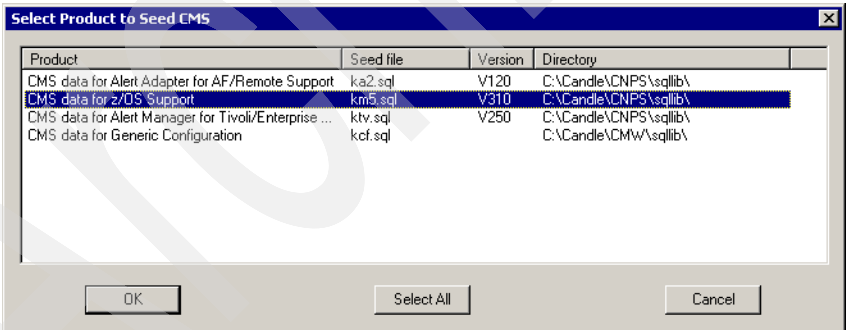


Figure 3-51   Seeding the z/OS

8. When the seeding is complete (this might take several minutes), the Seed Data Operation Complete window gives you information about seeding status and seed data location. Click **Save As** if you want to save the information in a text file. Click **Close** to close the window.

9. Stop and restart the hub Tivoli Enterprise Monitoring Server. In our case, we issue the command:

```
P CANSDSST
```

We restart it with:

```
S CANSDSST
```

10. Stop and restart the Tivoli Enterprise Portal Server from the Manage Candle Services window. You can use the  button.

3.4.5 Logging on to our Tivoli Enterprise Portal

To test our installation and configuration, we log on to our portal using the default sysadmin user ID as shown in Figure 3-52. The default user ID, sysadmin, has a blank password.

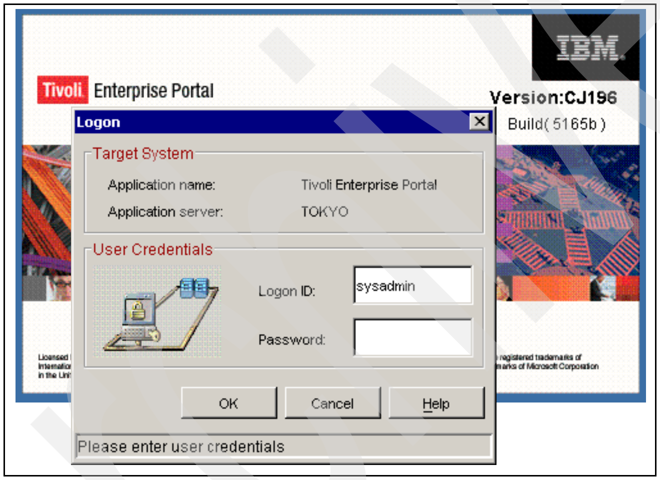


Figure 3-52 Logging on to Tivoli Enterprise Portal

We now see our z/OS sysplex details, and specifically our data from SC52 (Figure 3-53).

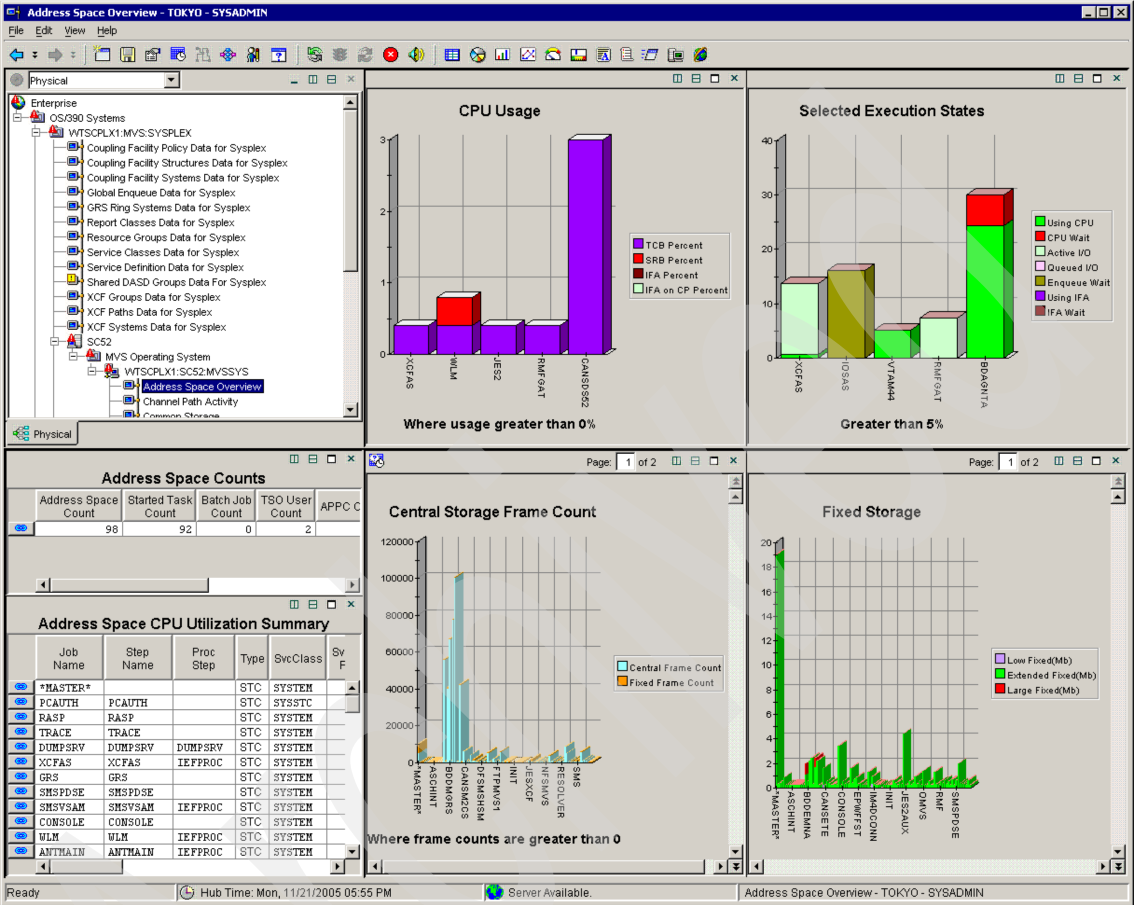


Figure 3-53 Completed Tivoli Enterprise Portal configuration

3.5 Moveable hub implementation

This section discusses the implementation of the moveable hub and the additional process that is required to be put in place to ensure that maintenance and seeding are up to date. To use a moveable hub, you must have a strict change management procedure in your enterprise. To see the moveable hub implementation document, search for the string `Moveable HUB` (case sensitive) at:

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

3.5.1 Moveable hub background

Hub Tivoli Enterprise Monitoring Server is the core part of IBM Tivoli OMEGAMON XE platform, which connects the front end of Tivoli Enterprise Portal and Tivoli Enterprise Portal Server and back-end remote Tivoli Enterprise Monitoring Servers and Tivoli Enterprise Monitoring Agents. If the hub Tivoli Enterprise Monitoring Server failed, what would happen? The IBM Tivoli OMEGAMON XE world would break in the middle with the front end unable to talk with the back end, even if the back end still works and data is still collected.

The z/OS platform has the concept of moveable hub. This enables restarting the hub Tivoli Enterprise Monitoring Server on a different member of the sysplex using a shared DASD. As the Tivoli Enterprise Monitoring Server is started on a different sysplex member, the clients (Candle Management Workstation and Tivoli Enterprise Portal Server) and remote Tivoli Enterprise Monitoring Servers must connect to the new hub, either by DNS mapping or using Dynamic Virtual IP addressing (DVIPA) support. Figure 3-54 shows the moveable hub concept.

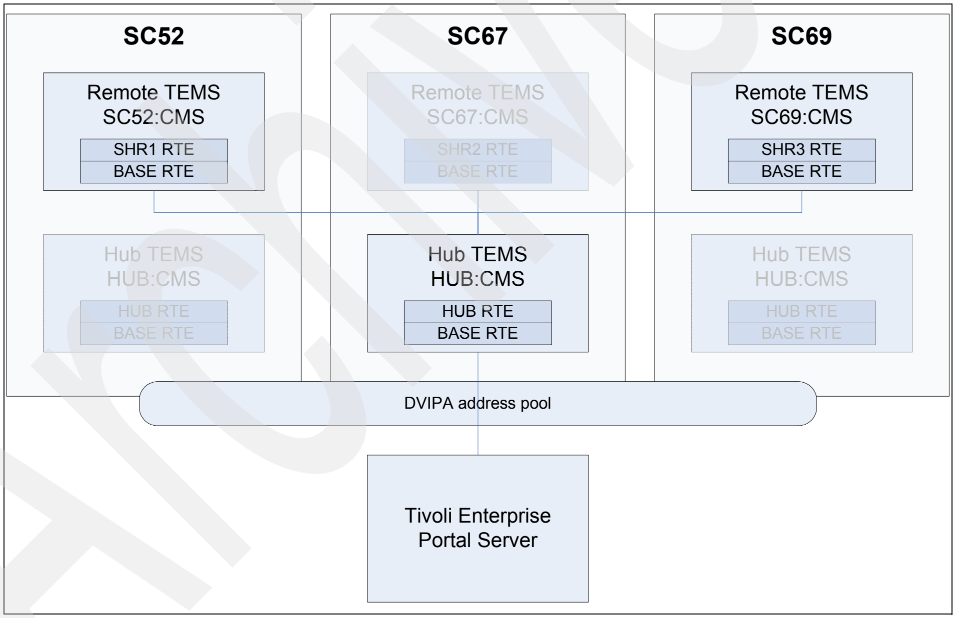


Figure 3-54 Moveable hub

The moveable hub defines the Tivoli Enterprise Monitoring Server on the primary sharing RTEs to be remote RTEs. We then create another sharing RTE that does not depend on a system-specific variable substitution (such as HUB RTE). This RTE can then be used on any member of the sysplex. The IP address that the hub Tivoli Enterprise Monitoring Server uses in the HUB RTE is not the physical

IP address of the machine; it is a DVIPA address that is taken from the DVIPA pool. The address in the DVIPA pool can be retrieved by any member of the sysplex when starting the hub Tivoli Enterprise Monitoring Server started task.

3.5.2 Creating a moveable hub

A hub Tivoli Enterprise Monitoring Server as configured by the configuration tool is by definition static. It cannot be moved from one LPAR to another without reconfiguring every component that would connect to it. The technique described here enables the Hub to be relocated to any suitable LPAR in the environment with no changes, and with minimal disruption to all components connecting to it.

Several areas must be considered on the moveable hub creation. Those are:

1. Defining the TCP/IP address for use with the moveable hub; typically this is involved with implementing DVIPA, see “Setting up TCP/IP access” on page 92. We assume that you are using DVIPA with TCP/IP.
2. Running the IBM Configuration Tool to define, build, configure, and load a new sharing RTE for the hub Tivoli Enterprise Monitoring Server. See “IBM Configuration Tool work” on page 93.
3. (optional) Configuring the RTE for the remote Tivoli Enterprise Monitoring Server. See “Remote Tivoli Enterprise Monitoring Server configuration” on page 94.
4. Working with a moveable hub requires strict change management to accommodate manual changes outside of the IBM Configuration Tool. See “Manual parameter member changes” on page 95.

Setting up TCP/IP access

The best option for a sysplex environment is to implement Dynamic Virtual IP Addressing (DVIPA). Otherwise, manual changes must be performed in KDCSSITE and KxxSHOST to list all candidate systems. The CT_CMSLIST for all Tivoli Enterprise Monitoring Agents must be expanded to include all IP addresses of all candidate systems.

The DVIPA takeover can be performed if you use the IP.PIPE protocol for hub Tivoli Enterprise Monitoring Server connectivity based on the started task name and the port used. The DVIPA information must be defined either using an OBEYFILE command or in the TCPIP PROFILE. Example 3-3 on page 93 shows a sample DVIPA definition to use port 1920 for a specific address, 9.12.6.27. We call the Tivoli Enterprise Monitoring Server started task CANSDHUB.

Example 3-3 DVIPA definition

```
VIPADYNAMIC
   VIPARANGE DEFINE 255.255.255.248 9.12.6.24
ENDVIPADYNAMIC

   . . .

PORT
 1920   TCP  CANSDHUB      BIND 9.12.6.27
```

As shown in Example 3-3, the DVIPA addresses are reserved using the VIPARANGE statement using the mask and the base address there. The binding to the address is determined using the port usage and started task name.

It is recommended that Tivoli Enterprise Monitoring Agents—both z/OS-based and distributed—connect to the remote Tivoli Enterprise Monitoring Server to avoid reconnection when the hub moved. This enables the Tivoli Enterprise Monitoring Agent to connect to a remote Tivoli Enterprise Monitoring Server location broker, instead of the global location broker with the hub Tivoli Enterprise Monitoring Server. Another approach uses an SNA-only connection for the Tivoli Enterprise Monitoring Agent. It is suggested that the concentrator hub for distributed Tivoli Enterprise Monitoring Agents' connections be directed to systems that are not hub candidates.

For a Tivoli Enterprise Monitoring Agent on z/OS that must run inside a Tivoli Enterprise Monitoring Server, such as CP, M3, OS, and S3; they should resides in remote Tivoli Enterprise Monitoring Server.

Reconnection to the hub Tivoli Enterprise Monitoring Server from a remote Tivoli Enterprise Monitoring Server or Tivoli Enterprise Monitoring Agent is performed using the CT_CMSLIST or KDCSSITE list of Tivoli Enterprise Monitoring Server. A reconnection retry will be performed automatically if the Tivoli Enterprise Monitoring Server connection goes down. The retry interval will be the same as the heartbeat time that is used by IBM Tivoli OMEGAMON XE. When the list is exhausted, it will start again from the top after waiting an interval equal to the heartbeat, which by default is 10 minutes.

IBM Configuration Tool work

Define in a new sharing RTE a new Tivoli Enterprise Monitoring Server. This new server should use the DVIPA-based TCP/IP address and specify a unique name for the alternate host name in the KDSENV variable KDCB0_HOSTNAME. You can also use a DNS entry specific for the DVIPA address. All communication parameters, such as node ID, VTAM names, and procedure name, should be unique and not system-specific names.

This new RTE will only be used to run the hub Tivoli Enterprise Monitoring Server.

Parameters that change from one LPAR to the next will be copied at runtime (see “Manual parameter member changes” on page 95).

Different products require different actions to be performed, such as:

- For sysplex products, such as OE, CP, DP, and IP, you must perform only the “Register with local CMS” step for each product.
- For M2 and MQ, perform “Register with local CMS.” For MC, perform “Register with local CMS” and “Configure Historical Datasets.” If DF is being configured, do not run job DF#3HUB. If it was run, delete all KDF* members from RKANPAR except for KDFLLIST, KDFLLISD, and KDFPDICT. All members will have been created in the LPAR RTE and will come from the concatenated local RKANPAR.
- For the AM product, perform only “Install product provided solutions into the CMS.”
- For IBM Tivoli OMEGAMON XE on z/OS, you must place the RTE in a sysplex. The hub Tivoli Enterprise Monitoring Server should not be considered for sysplex proxy duties in a large sysplex. There should be at least one proxy candidate left in the sysplex after the target remote has been replaced by the hub. Perform “Exclude from Plex-Proxy Rotation.” This can be done only if other RTEs already exist in the sysplex; define LPAR RTEs before the HUB and use the Advanced options to point the remote Tivoli Enterprise Monitoring Server to the not-yet-defined HUB (as if it were a distributed HUB).

On systems that are candidates for moveable hub, standalone agents can be configured to find an alternate Tivoli Enterprise Monitoring Server when their local Tivoli Enterprise Monitoring Server is being replaced by the hub. This can be done in the IBM Configuration Tool by specifying the hub as a secondary Tivoli Enterprise Monitoring Server, which ensures that the agent will always be able to connect to a Tivoli Enterprise Monitoring Server.

When configuring a moveable hub in a test or small environment, the hub can also be configured as primary sysplex proxy. Then you must perform the “Configure Historical Datasets” step. There are additional parameter members in RKDSPARM to be copied, such as KOSAL and KOSPG.

Remote Tivoli Enterprise Monitoring Server configuration

For each sysplex member, the sharing RTE must be configured as a remote Tivoli Enterprise Monitoring Server with all of the necessary products. Connect these remote Tivoli Enterprise Monitoring Servers to the moveable hub that we defined in “IBM Configuration Tool work” on page 93 with a unique Tivoli

Enterprise Monitoring Server name. Run the RKANSAM(KDSDREM) job to remove RKDS* datasets not used by remote CMSs.

Manual parameter member changes

To accommodate the moveable hub, certain actions must be performed outside of the IBM Configuration Tool. Typically this includes modifying the Tivoli Enterprise Monitoring Server and defining a new dataset for holding customized system specific parameters. We call this dataset HUBPRMS.

> **Note:** It is important to perform a strict change management for these changes as any maintenance or update to the product that requires a rebuild of RTE will override these manual changes. The process should accommodate at least the following requirements:
>
> - Members that are modified in RKANPAR
> - Changes to Tivoli Enterprise Monitoring Server started tasks
> - Members copied or modified in the HUBPRMS dataset

Members that must be modified in RKANPARM are:

- KDCSSITE: In all RTEs, KDCSSITE must include all IP.PIPE specifications for each candidate system. See Example 3-4.

Example 3-4 Sample KDCSSITE

```
SNA:&SVVSNTID..K&SYSCLONE.DSLB.CANCTDCS.SNASOCKETS
IP:wtschub,wtsc52,wtsc67,wtsc69
IP.PIPE:wtschub,wtsc52,wtsc67,wtsc69
```

- KxxENV: The CT_CMSLIST in KxxENV for all components that connect directly to the hub must be modified as shown in Example 3-5. However, it is recommended that all components report to their respective remote Tivoli Enterprise Monitoring Server running locally.

Example 3-5 Sample KxxENV

```
CT_CMSLIST=\
   IPE:wtsc52;\
   IP:wtsc67;\
   IP:wtsc69;\
   IP.PIPE:wtsc52;\
   IP.PIPE:wtsc67;\
   IP.PIPE:wtsc69;
```

- KxxSHOST members must contain the appropriate entries to resolve all host names that are added, similar to Example 3-6. The candidate lines are added manually.

Example 3-6 Sample KxxSHOST member

```
SNA:USIBMSC.K52DSLB.CANCTDCS.SNASOCKETS
IP:wtsc52
IPPIPE:wtsc52
IP:wtsc67
IPPIPE:wtsc67
IP:wtsc69
IPPIPE:wtsc69
```

Further modification is needed: Some parameter members require information that must be modified to contain system-specific information. This means that you must pre-define the system-specific information in a dataset that would be copied over the parameter members that are used in execution of the hub or the remote Tivoli Enterprise Monitoring Server.

The parameter members that must be modified before running the hub Tivoli Enterprise Monitoring Server are:

- KDSENV: The CMS_NODEID parameter is used to find the Tivoli Enterprise Monitoring Server. There will be two copies for KDSENV per system, one for acting as the hub, the other for remote.
- KLXINTCP: May have to be modified to accommodate system-specific TCP/IP address space differences.
- KEPOPTN: The REMOTEID() parameter is used by EPILOG to find the Tivoli Enterprise Monitoring Server. There will be two copies for KEPOPTN per system, one for acting as the hub, the other for remote.
- KM2IPARM: IBM Tivoli OMEGAMON XE on z/OS parameter.

To perform this, a pre-step is introduced in the Tivoli Enterprise Monitoring Server started task to copy members from HUBPRMS to the appropriate RKANPAR datasets. Figure 3-55 illustrates this process.

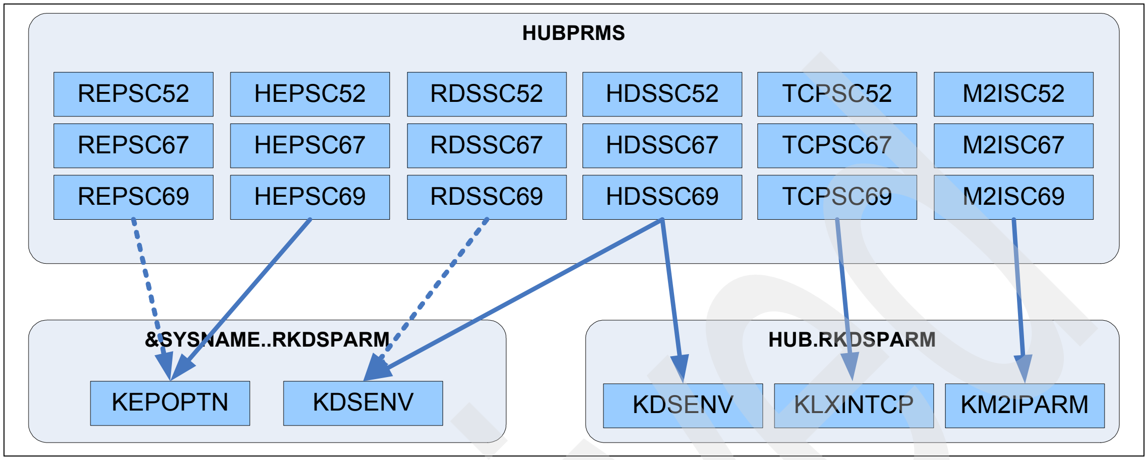


Figure 3-55 Parameter copying scheme

Example 3-7 shows the sample started task for the hub Tivoli Enterprise Monitoring Server.

Example 3-7 Hub Tivoli Enterprise Monitoring Server started task

```
//* give the HUB the right KDSENV and KDCSHOST, fix M2/EP parms
//* note that HUB.HUBCMS and HUB.RKANPAR could be the same dataset
//*
//COPYPRM EXEC PGM=IEBCOPY
//SYSPRINT DD  SYSOUT=*
//RKANPARH DD  DSN=&RHILEV..HUB.RKANPAR,DISP=SHR
//RKANPAR  DD  DSN=&RHILEV..CCC&SYSNAME..RKANPAR,DISP=SHR
//HUBPRMS  DD  DSN=&RHILEV..HUB.HUBPRMS,DISP=SHR
//SYSIN    DD  DSN=&RHILEV..HUB.HUBPRMS(CPYH&SYSNAME.),DISP=SHR
//*
//*
//CMS     EXEC  PGM=KLV…,COND=(0,LT)
. . . .
//*     HUB RTE product members
//*     other members specific to this LPAR (KDF*)
//*
//RKANCMD DD DSN=&RHILEV..HUB.RKANCMD,DISP=SHR
//        DD DSN=&RHILEV..CCC&SYSNAME..RKANCMD,DISP=SHR
//*
//*     HUB RTE product parms
```

```
//*     M2 and PDS management parms specific to this LPAR
//*
//RKANPAR DD DSN=&RHILEV..HUB.RKANPAR,DISP=SHR
//        DD DSN=&RHILEV..CCC&SYSNAME..RKANPAR,DISP=SHR
. . . .
```

The CPYH&SYSNAME member contains IEBCOPY control statements that are used to move the members into their appropriate location. Example 3-8 shows the sample copy control statement for SC52 hub. Compressing the dataset prior to copying is not needed if you are using PDSE.

Example 3-8 CPYHSC52 IEBCOPY statements

```
 COPY INDD=RKANPARH,OUTDD=RKANPARH  ** unless dataset is PDSE **
 COPY INDD=HUBPRMS,OUTDD=RKANPARH
   S M=((HDSSC52,KDSENV,R))
   S M=((TCPSC52,KLXINTCP,R))
   S M=((KM2ISC52,KM2IPARM,R))
 COPY INDD=RKANPAR,OUTDD=RKANPAR  ** unless dataset is PDSE **
 COPY INDD=HUBPRMS,OUTDD=RKANPAR
   S M=((HDSSC52,KDSENV,R))
   S M=((HEPSC52,KEPOPTN,R))  ** if EPILOG is being used **
```

The remote Tivoli Enterprise Monitoring Server started task also must be modified to accommodate this copying step as shown in Example 3-9.

Example 3-9 Remote Tivoli Enterprise Monitoring Server started task

```
//COPYPRM EXEC PGM=IEBCOPY
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  DSN=&RHILEV..HUB.HUBCMS(CPYR&SYSNAME.),DISP=SHR
//RKANPAR  DD  DSN=&RHILEV..CCC&SYSNAME..RKANPAR,DISP=SHR
//HUBPRMS  DD  DSN=&RHILEV..HUB.HUBPRMS,DISP=SHR
//*
//CMS     EXEC  PGM=KLV…
. . . .
```

The CPYRxxxx member contains IEBCOPY control statements (Example 3-10).

Example 3-10 CPYRSC52 member

```
 COPY INDD=RKANPAR,OUTDD=RKANPAR  ** unless dataset is PDSE **
 COPY INDD=HUBPRMS,OUTDD=RKANPAR
   S M=((RDSSC52,KDSENV,R))
   S M=((REPSC52,KEPOPTN,R))  ** if EPILOG is being used **
```

If some products that are running inside the Tivoli Enterprise Monitoring Server are configured and expected to collect historical data, KPDCTL, KPDCTL2, and KPDDEFIN may have to be modified for each system, similar to the process provided in Figure 3-55 on page 97. Combined content for an original hub and remote parameter for each system must be copied to the hub's parameter dataset. Figure 3-56 shows the combination.

Figure 3-56 Historical data control members

After preparing all system-specific members, the IEBCOPY control statements for the hub Tivoli Enterprise Monitoring Server described in Example 3-8 on page 98 can be modified as shown in Example 3-11.

Example 3-11 Sample copy statements.

```
COPY INDD=RKANPARH,OUTDD=RKANPARH  ** unless dataset is PDSE **
COPY INDD=HUBPRMS,OUTDD=RKANPARH
  S M=((HDSSC52,KDSENV,R))
  S M=((TCPSC52,KLXINTCP,R))
  S M=((KM2ISC52,KM2IPARM,R))
  S M=((CTLSC52,KPDCTL,R))
  S M=((CTL2SC52,KPDCTL2,R))
  S M=((DEFISC52,KPDDEFIN,R))
COPY INDD=RKANPAR,OUTDD=RKANPAR  ** unless dataset is PDSE **
COPY INDD=HUBPRMS,OUTDD=RKANPAR
  S M=((HDSSC52,KDSENV,R))
  S M=((HEPSC52,KEPOPTN,R))  ** if EPILOG is being used **
COPY INDD=HUBPRMS,OUTDD=RKANPARH
```

For products that run in standalone agent address spaces, historical data collection should be configured at the agent only.

3.6 Using IBM Tivoli Monitoring V6.1

This section discusses the implementation of IBM Tivoli Monitoring V6.1 in Linux on zSeries to act the as hub Tivoli Enterprise Monitoring Server and Tivoli Enterprise Portal Server. The discussion in this section includes:

- 3.6.1, “Prerequisites for Linux on zSeries server” on page 100
- 3.6.2, “Installation of IBM Tivoli Monitoring” on page 101
- 3.6.3, “Populating application support modules” on page 108

3.6.1 Prerequisites for Linux on zSeries server

Supported Linux on zSeries server platforms are:

- Red Hat Enterprise Linux (RHEL) Version 3.0 or 4.0
- SUSE Linux Enterprise Server (SLES) Version 8.0 or 9.0

Our setup used SLES 9 installed as a VM guest. We allocated these file systems:

- /opt/IBM: The default installation directory for IBM products in Linux on zSeries

- /DB2: The home directory for db2inst1 user; we created the db2inst1 user manually to point to the /DB2 directory as the home directory.

> **Note:** The db2inst1 user ID must have a primary group other than users. We created the db2iadm1 group as the primary group for db2inst1, and modify /etc/passwd to change the db2inst1 home directory and primary group.

We pre-installed DB2 using a typical installation. The db2inst1 instance is created with the default configuration.

3.6.2 Installation of IBM Tivoli Monitoring

IBM Tivoli Monitoring V6 is installed using the installation CD image transferred to Linux on zSeries. Start the installation using the **`install.sh`** command from the installation CD. The installation process is performed using a command line interface. Example 3-12 shows the invocation. We installed the product in /opt/IBM/ITM. Our responses are shown in bold.

Example 3-12 Installation dialog for IBM Tivoli Monitoring (part 1 of 6)

```
linux2:/opt/IBM/itm61code # ./install.sh

Enter the name of the IBM Tivoli Monitoring directory
[ default = /opt/IBM/ITM ]:

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

Select one of the following:

1) Install products to the local host.
2) Install products to depot for remote deployment (requires TEMS).
3) Exit install.

Please enter a valid number:  1
```

After you select the language and accept the license agreement, the dialog continues as in Example 3-13. The installation uses IBM Global Security Kit to provide encryption. Default encryption key is `IBMTivoliMonitoringEncryptionKey`.

Example 3-13 Installation dialog for IBM Tivoli Monitoring (part 2 of 6)

```
Preparing to install the IBM Global Security Kit (GSkit)
Preparing packages for installation...
```

```
gsk7bas-7.0-3.18
 Will enable automatic agent initiation after reboot.

Enter a 32-character encryption key, or just press Enter to use the
default
        Default = IBMTivoliMonitoringEncryptionKey
....+....1....+....2....+....3..
GSkit encryption key has been set.
Key File directory: /opt/IBM/ITM/keyfiles
```

The next part selects the operating system platform to install to. If you run on a supported Linux on zSeries platform, the default operating system selection should have been selected as shown in Example 3-14.

Example 3-14 Installation dialog for IBM Tivoli Monitoring (part 3 of 6)

```
Product packages are available in /opt/IBM/itm61code/unix

Product packages are available for the following operating systems and
component support categories:

 1) Linux S390 R2.4 (32 bit)
 2) Linux S390 R2.4 (64 bit)
 3) Linux S390 R2.4 GCC 2.9.5 (64 bit)
 4) Linux S390 R2.6 (32 bit)
 5) Linux S390 R2.6 (64 bit)
 6) Linux S390 R2.6 GCC 2.9.5 (32 bit)
 7) Linux S390 R2.6 GCC 2.9.5 (64 bit)
 8) Tivoli Enterprise Portal Browser Client support
 9) Tivoli Enterprise Portal Server support

Type the number for the OS or component support category you want, or type
"q" to quit selection
[ number "4" or "Linux S390 R2.6 (32 bit)" is default ]:

You selected number "4" or "Linux S390 R2.6 (32 bit)"

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

As shown in Example 3-15 on page 103, we install all components. You need to install at least the Tivoli Enterprise Monitoring Server V06.10.00.00 and Tivoli Enterprise Portal Server V06.10.00.00 depending on your configuration. These components can reside on one or more machines.

Example 3-15 Installation dialog for IBM Tivoli Monitoring (part 4 of 6)

```
The following products are available for installation:

 1) IBM Eclipse Help Server  V06.10.00.00
 2) Monitoring Agent for Linux OS  V06.10.00.00
 3) Monitoring Agent for UNIX Logs  V06.10.00.00
 4) Tivoli Enterprise Monitoring Server  V06.10.00.00
 5) Tivoli Enterprise Portal Server  V06.10.00.00
 6) Tivoli Enterprise Services User Interface  V06.10.00.00
 7) Universal Agent  V06.10.00.00
 8) all of the above

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

Type your selections here:  8
```

Next shows the installation progress messages. We enter the Tivoli Enterprise Monitoring Server name as shown in Example 3-16.

Example 3-16 Installation dialog for IBM Tivoli Monitoring (part 5 of 6)

```
 ... installing "IBM Eclipse Help Server  V06.10.00.00 for Linux S390
R2.6 (32 bit)"; please wait.

 => installed "IBM Eclipse Help Server  V06.10.00.00 for Linux S390
R2.6 (32 bit)."
 ... Initializing database for IBM Eclipse Help Server  V06.10.00.00
for Linux S390 R2.6 (32 bit).
 ... IBM Eclipse Help Server  V06.10.00.00 for Linux S390 R2.6 (32 bit)
initialized.

 ... installing "Monitoring Agent for Linux OS  V06.10.00.00 for Linux
S390 R2.6 (32 bit)"; please wait.

 => installed "Monitoring Agent for Linux OS  V06.10.00.00 for Linux
S390 R2.6 (32 bit)."
 ... Initializing database for Monitoring Agent for Linux OS
V06.10.00.00 for Linux S390 R2.6 (32 bit).
 ... Monitoring Agent for Linux OS  V06.10.00.00 for Linux S390 R2.6
(32 bit) initialized.
```

```
 ... installing "Monitoring Agent for UNIX Logs  V06.10.00.00 for Linux
S390 R2.6 (32 bit)"; please wait.

 => installed "Monitoring Agent for UNIX Logs  V06.10.00.00 for Linux
S390 R2.6 (32 bit)."
 ... Initializing database for Monitoring Agent for UNIX Logs
V06.10.00.00 for Linux S390 R2.6 (32 bit).
 ... Monitoring Agent for UNIX Logs  V06.10.00.00 for Linux S390 R2.6
(32 bit) initialized.

 ... installing "Tivoli Enterprise Monitoring Server  V06.10.00.00 for
Linux S390 R2.6 (32 bit)"; please wait.

 => installed "Tivoli Enterprise Monitoring Server  V06.10.00.00 for
Linux S390 R2.6 (32 bit)."
 ... Initializing database for Tivoli Enterprise Monitoring Server
V06.10.00.00 for Linux S390 R2.6 (32 bit).
Please enter TEMS name: LINUX2:CMS
... creating config file
"/opt/IBM/ITM/config/linux2_ms_LINUX2:CMS.config"
... creating file "/opt/IBM/ITM/tables/LINUX2:CMS/glb_site.txt."
... updating "/opt/IBM/ITM/config/kbbenv"
... verifying Hot Standby.
 ... Tivoli Enterprise Monitoring Server  V06.10.00.00 for Linux S390
R2.6 (32 bit) initialized.

 ... installing "Tivoli Enterprise Portal Server  V06.10.00.00 for
Linux S390 R2.6 (32 bit)"; please wait.
 ... running InstallPresentation.sh
 ... InstallPresentation.sh completion code = 126
 ... Tivoli Enterprise Portal Server  V06.10.00.00 for Linux S390 R2.6
(32 bit) initialized.

 ... installing "Tivoli Enterprise Services User Interface
V06.10.00.00 for Linux S390 R2.6 (32 bit)"; please wait.

 => installed "Tivoli Enterprise Services User Interface  V06.10.00.00
for Linux S390 R2.6 (32 bit)."
 ... Initializing database for Tivoli Enterprise Services User
Interface  V06.10.00.00 for Linux S390 R2.6 (32 bit).
 ... Tivoli Enterprise Services User Interface  V06.10.00.00 for Linux
S390 R2.6 (32 bit) initialized.

 ... installing "Universal Agent  V06.10.00.00 for Linux S390 R2.6 (32
bit)"; please wait.
```

```
 => installed "Universal Agent  V06.10.00.00 for Linux S390 R2.6 (32
bit)."
 ... Initializing database for Universal Agent  V06.10.00.00 for Linux
S390 R2.6 (32 bit).
 ... Universal Agent  V06.10.00.00 for Linux S390 R2.6 (32 bit)
initialized.
```

The last part configures Tivoli Enterprise Portal Server. (If you have any application support files, we install application support files in 3.6.3, "Populating application support modules" on page 108.) Example 3-17 shows the configuration of the Tivoli Enterprise Portal Server.

Example 3-17 Installation dialog for IBM Tivoli Monitoring (part 6 of 6)

```
If you are installing Tivoli Enterprise Portal Server (TEPS) or Tivoli
Enterprise Portal Desktop Client (TEP) for the first time you will
probably want to install product support to the TEPS and TEP for the
agent products which you plan to use.  This gives you product specific
function within the TEP.  To install support packages choose yes below
or run the install again at a later time and when prompted to choose an
operating system or component
support category choose the appropriate support category.

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


 ... postprocessing; please wait.
 ... Copying work files, please wait.

 ... finished postprocessing.

 Installation step complete.

 As a reminder, you should install product support on each of your TEM
 servers for any agents you have just installed. This is done via the
 "[ITM home]/bin/itmcmd support" command on your TEM servers.

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

The next step is to configure Tivoli Enterprise Monitoring Server. Configuration is performed using the **`itmcmd config`** command. You can configure this using a graphical interface similar to Manage Candle Services window using the command **CandleManage** or **itmcmd manage**. Example 3-18 shows the dialog.

Example 3-18 Configuring Tivoli Enterprise Monitoring Server

```
linux2:/opt/IBM/ITM/bin # ./itmcmd config -S -t LINUX2:CMS
Configuring TEMS...

Hub or Remote [*LOCAL or *REMOTE] (Default is: *LOCAL):
TEMS Host Name (Default is: linux2):

Network Protocol 1 [ip, sna, ip.pipe, or ip.spipe] (Default is:
ip.pipe):

     Now choose the next protocol from one of these:
     - ip
     - sna
     - ip.spipe
     - none
Network Protocol 2 (Default is: none): none
IP.PIPE Port Number (Default is: 1918):
Enter name of KDC_PARTITION (Default is: null):
Enter path and name of KDC_PARTITIONFILE (Default is:
/opt/IBM/ITM/tables/LINUX2:CMS/partition.txt):

Configuration Auditing? [YES or NO] (Default is: YES): NO
Hot Standby TEMS Host Name (Default is: none):
Enter Optional Primary Network Name or "none" :(Default is: none):
Security: Validate User ? [YES or NO] (Default is: NO):

TEC Event Integration Facility? (Default is: NO):
Disable Workflow Policy/Tivoli Emitter Agent Event Forwarding? (Default
is: NO):
 ... Writing to database file for ms.

************************

Editor for SOAP hubs list

************************

Hubs
##      CMS_Name
1       ip.pipe:LINUX2:CMS[1918]
```

```
A)dd, R)emove ##, M)odify Hub ##, U)serAccess ##, C)ancel, S)ave/exit:
S
... creating config file
"/opt/IBM/ITM/config/linux2_ms_LINUX2:CMS.config"
... creating file "/opt/IBM/ITM/tables/LINUX2:CMS/glb_site.txt."
... updating "/opt/IBM/ITM/config/kbbenv"
... verifying Hot Standby.
CMS configuration completed...
```

The Tivoli Enterprise Monitoring Server that we configure is a local or hub Tivoli Enterprise Monitoring Server with the name of LINUX2:CMS. It only listens to IP:PIPE communication at port 1918.

> **Attention:** Do *not* define SNA protocol for the Tivoli Enterprise Monitoring Server if your Linux on zSeries is not configured with one.

Tivoli Enterprise Portal Server must then be configured. This includes populating and configuring the database function. This configuration also uses the **itmcmd config** command as shown in Example 3-19.

Example 3-19 Configuring the portal server

```
linux2:/opt/IBM/ITM/bin # ./itmcmd config -A cq

Agent configuration started...

Will this agent connect to a CMS? [YES or NO] (Default is: YES): YES
CMS Host Name (Default is: 9.12.4.112): 9.12.4.112

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

     Now choose the next protocol from one of these:
     - ip
     - sna
     - ip.spipe
     - none
Network Protocol 2 (Default is: none): none
IP.PIPE Port Number (Default is: 1918): 1918
Enter name of KDC_PARTITION (Default is: null):

Enter Optional Primary Network Name or "none" (Default is: none): none
Enable SSL for TEP Clients (y/n)(Default is: N): N
Enter the DB2 instance name(Default is: db2inst1): db2inst1
```

```
Enter the DB2 admin ID(Default is: db2inst1): db2inst1
Enter the password for the DB2 admin ID(Default is: ******):
Re-type: Enter the password for the DB2 admin ID(Default is: ******):
Enter the TEPS DB2 database name(Default is: TEPS): TEPS
Enter the TEPS DB2 database login ID(Default is: itmuser): itmuser
Enter the password for the TEPS DB2 database login ID(Default is:
******):
Re-type: Enter the password for the TEPS DB2 database login ID(Default
is: ******):
Is it OK to create the TEPS login ID if not found (y/n)(Default is: Y):
Y
Enter the Warehouse database name(Default is: WAREHOUS): WAREHOUS
Enter the Warehouse user ID(Default is: itmuser): itmuser
Enter the password for the Warehouse user ID(Default is: ******):
Re-type: Enter the password for the Warehouse user ID(Default is:
******):
 ... running InstallPresentation.sh
 ... InstallPresentation.sh completed
Agent configuration completed...
```

We can now start the Tivoli Enterprise Monitoring Server and Tivoli Enterprise Portal Server. Tivoli Enterprise Monitoring Server is started using the command **itmcmd server start LINUX2:CMS** and Tivoli Enterprise Portal Server is started using the command **itmcmd agent start cq**.

Note: The Tivoli Enterprise Portal Server is considered an agent with the product code of cq.

3.6.3 Populating application support modules

The application support modules are installed using the enablement CD for OMEGAMON LCD7-0817. This CD has the graphical interface to implement z/OS-based products on an IBM Tivoli Monitoring V6 environment.

For a Linux on zSeries environment, start the wizard using the command **setuplinux390.bin**. You must have a running X Window System server and point the DISPLAY environment variable to that X Windows server.

1. The first dialog shows the welcome window (Figure 3-57). Click **Next** to proceed.

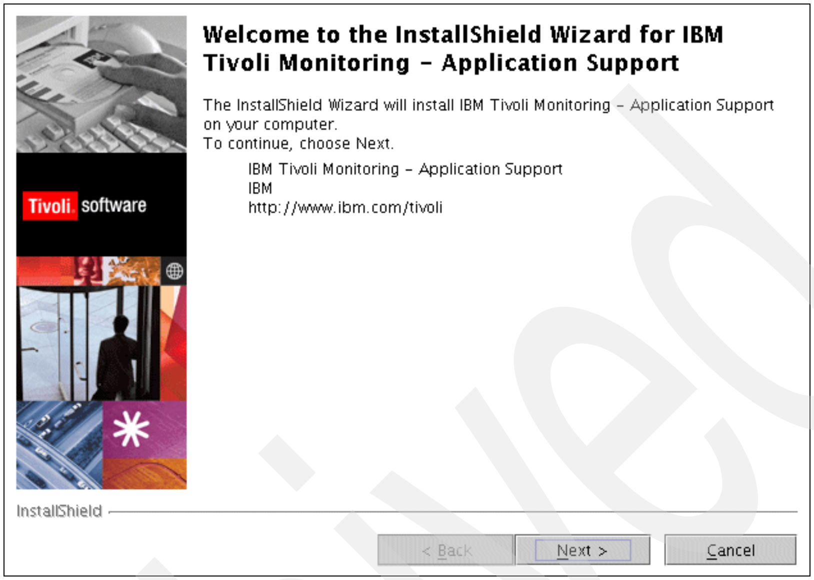


Figure 3-57 Welcome window

2. Specify the installation source and target as shown in Figure 3-58.

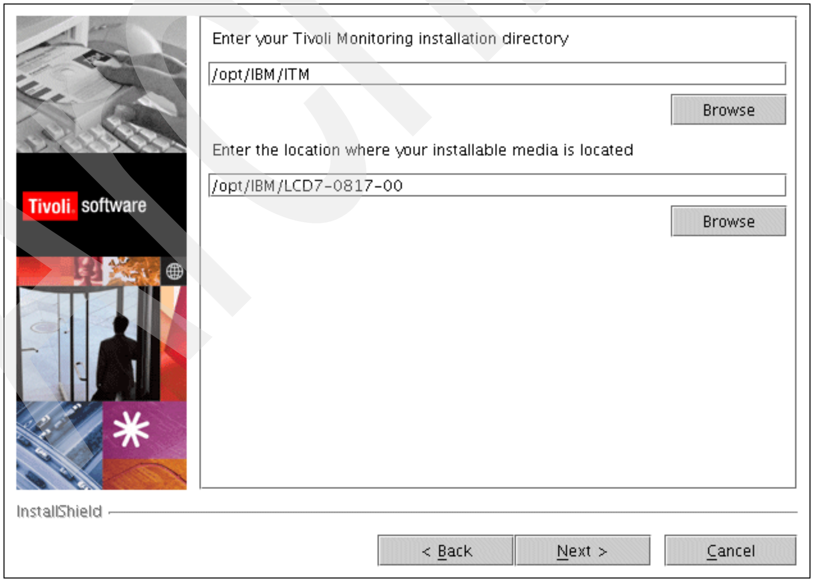


Figure 3-58 Installation paths

3. Specify the configuration action. Depending on the feature that is installed on the machine, you can select either or both of the options in Figure 3-59.

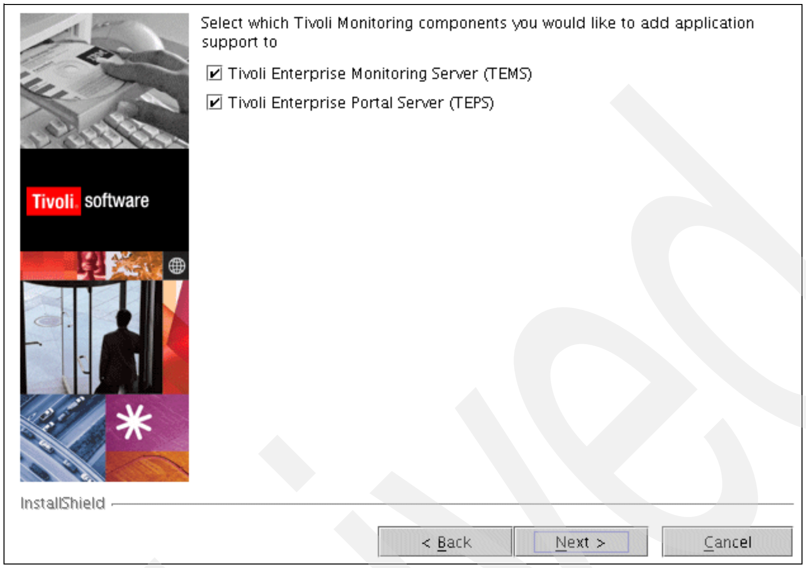


Figure 3-59 Installation options

4. Select the products that you want to install support for. Some options are mutually exclusive in that they are disabled when you select some of them.

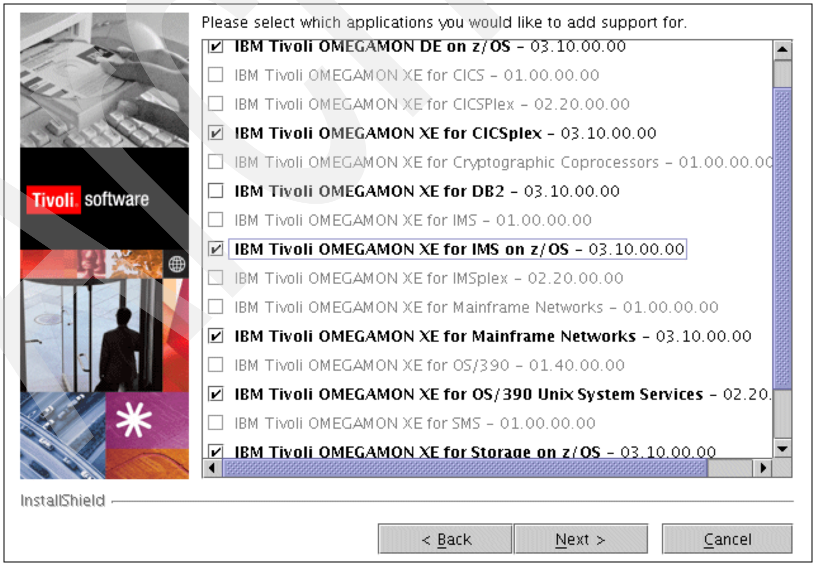


Figure 3-60 Available components

5. If you select OMEGAMON DE, confirm that you have the license to run it. Otherwise, confirm the installation steps (Figure 3-61) and proceed.

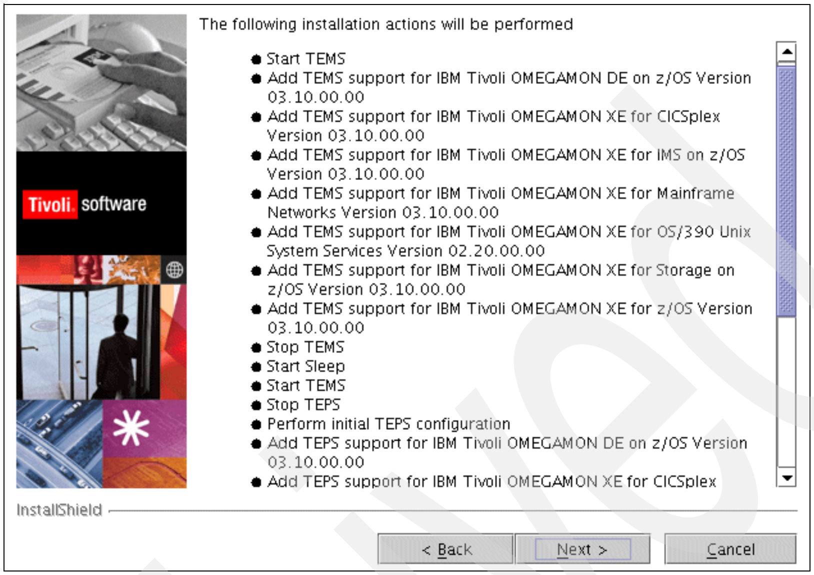


Figure 3-61 Installation step summary

6. Figure 3-62 shows that the installation completed successfully.

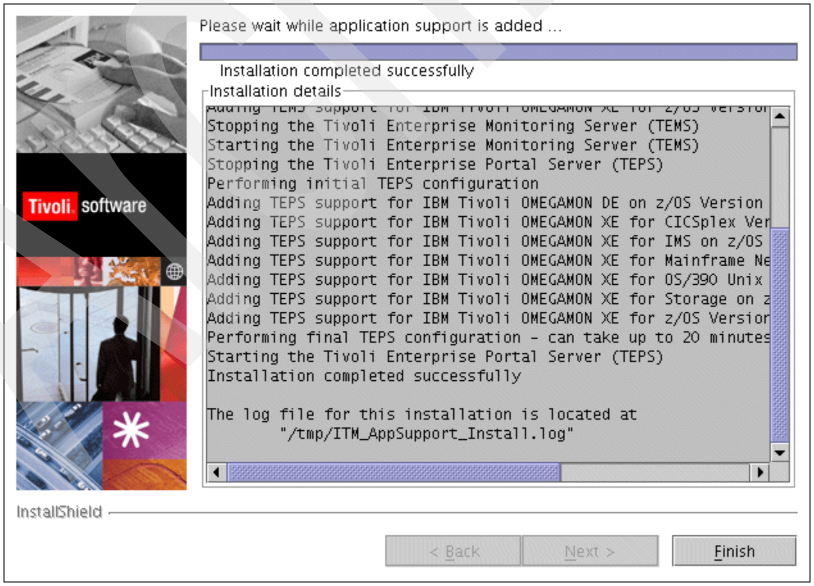


Figure 3-62 Installation success

4

Maintenance and problem determination

In this chapter, we explain some of the procedures and methods for performing maintenance and problem determination. The topics in this section are:

4.1 Implementing z/OS maintenance

This section discusses applying maintenance of OMEGAMON products running on the z/OS platform.

4.1.1 PSP bucket

Before upgrading or installing maintenance, it is recommended that you review the appropriate Preventive Service Planning (PSP) for the OMEGAMON product you are installing. The OMEGAMON PSP bucket information can be found at:

`https://techsupport.services.ibm.com/server/390.psp390`

You need an IBM user ID to access this information. After you sign on, enter `OMEGAMON` in the search field. Information based on upgrade ID and subset ID appear; each one provides a short description of the subset. For example, the PSP bucket for IBM Tivoli OMEGAMON XE for CICS on z/OS V3.1.0 shows the components with upgrade ID OMXEC5310 in Table 4-1.

Table 4-1 List of OMEGAMON for CICS PSP buckets

Upgrade ID	Subset	Description
OMXEC5310	HKC5310	OMEGAMON for CICS on z/OS
OMXEC5310	HKDS360	MGMNT SERVER on z/OS V. 3.6.0
OMXEC5310	HKET550	ITMS:Engine
OMXEC5310	HKLV190	END-TO-END
OMXEC5310	HKOB550	OMNIMON BASE
OMXEC5310	HKCI310	INSTALL CONFIG ASSISTANCE TOOL
OMXEC5310	HKC2550	OMEGAMON II FOR CICS V. 5.5.0

4.1.2 Working with PTFs and fixes

Applying an individual PTF or fix requires the following steps:

1. Download the fix from the IBM support site. (See “Getting PTFs from IBM” on page 115.)
2. Perform SMP/E processing to load the fix into target and distribution libraries, as discussed in “SMP/E operation” on page 117.
3. Run the configuration tool to reload the runtime environment as explained in “Reload RTE using the configuration tool” on page 117.

Getting PTFs from IBM

For zSeries-based products, IBM PTFs can be downloaded from ShopzSeries or the IBM Support page. The recommended download process can be found at:

`http://www.ibm.com/support/docview.wss?uid=swg21225816&rs=2271`

The following procedure is for downloading a PTF from IBM:

1. At `http://www.ibm.com`, select **Support & downloads** in the navigation bar.
2. On the Support & downloads page, select **Software** and then select **Tivoli**. Select the product you want to obtain fixes for.
3. Under Self help, in the Solve a problem section, click **APARs**. APAR is an abbreviation of authorized program analysis report. You can further limit the result by entering a search term.
4. On the Search results page scroll down to find the particular APAR that you are interested in. Click the APAR title to get into the APAR content.
5. Get the appropriate Problem Temporary Fix (PTF) under the Fix information as shown in Figure 4-1.

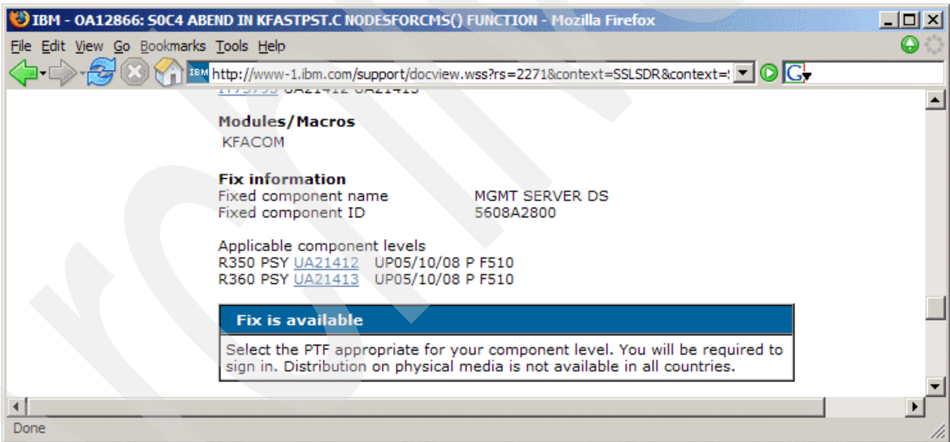


Figure 4-1 Link to the PTF

6. On the Get zSeries–related fixes, there are several ways to order the PTFs. You can use the Electronic delivery link to download the fix. (You may need to sign in using your IBM ID. You have to register the ID if you do not have one.)
7. On the Download selected fixes page under Packaging options, check **Yes** for all options as shown in Figure 4-2 on page 116. Validate your e-mail address, as this is where the PTF will be sent. Click **Continue**.

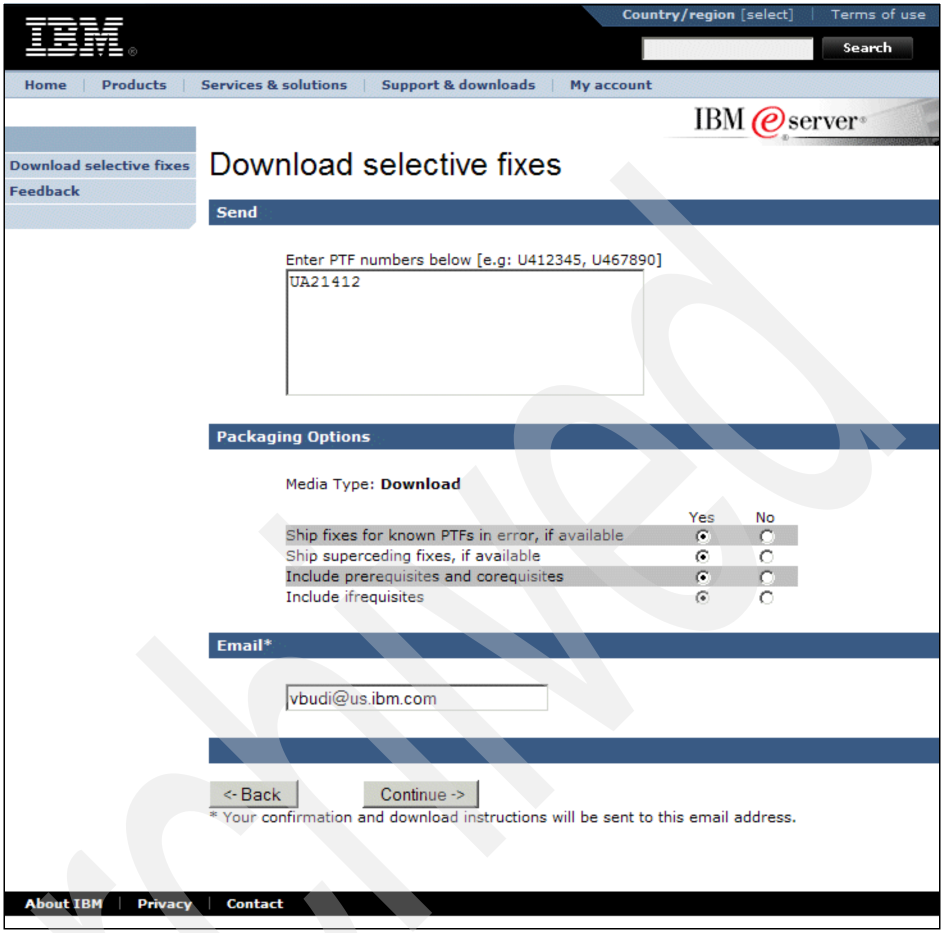


Figure 4-2 PTF download option

8. Verify the order information and click **Submit**. This sends the order for the PTF to the e-mail address you provided. You will get several e-mails:
   - Confirmation of your order
   - Confirmation that the order is sent and download instructions
9. The PTF will be made available on a public FTP site. It contains several files, typically:
   - FTP instruction
   - Packing list and hold data information
   - Binary of the PTF
10. The PTF must be saved to a z/OS dataset with fixed block and record length of 1024 (FB 1024) as it is in a tersed data format.

SMP/E operation

After you get the PTF information, load the PTF into SMP/E using the RECEIVE - APPLY - ACCEPT method:

1. Unterse the PTF or PTFs from "Getting PTFs from IBM" on page 115.
2. Receive the PTF into the same CSI that was used for the original product. You must perform the receive function using a JCL or SMP/E dialog.
3. Perform Apply Check the PTF(s) and then Apply the PTF(s).
4. Perform Accept Check the PTF(s) and the Accept the PTF(s).

> **Note:** The JCL for SMP/E operation for a PTF is an external JCL. It is not generated by the configuration tool. Some implementations choose to ACCEPT the PTFs only before implementing the next maintenance package instead of immediately after APPLY.

Reload RTE using the configuration tool

After the PTF is loaded into the target library, you can use the configuration tool to reload the runtime environment:

1. Start the configuration tool from TSO by issuing the **`EXEC &shilev.INSTLIB`** command, where `shilev` is the high-level qualifier of the OMEGAMON configuration tool.
2. Select option `3` - Configure products.
3. Select option `2` to configure product and then select a product to configure.
4. On the RUNTIME ENVIRONMENTS (RTEs) panel, place an `L` in front of the appropriate RTE name (or names).
5. Submit the generated JCL, and the fix is applied and loaded.

4.2 Distributed environment fix packs

Different from the z/OS environment, the distributed environment implements fixes in fix packs. This set of fixes are packaged and distributed in a single media. Fix packs typically correspond to a set of z/OS PTFs to work properly.

You can retrieve the fix pack information from the IBM Web site and download the fixpack with the following procedure:

1. At `http://www.ibm.com`, click **Support & Downloads** and select **Download**.
2. Find the product name, such as IBM Tivoli Management Portal for Distributed Systems. A window will show available fixpacks (Figure 4-3 on page 118).

Select a specific fixpack, such as 1.9.6-TIV-MgtPortal-FP0003, which means fixpack 3 of IBM Tivoli Management Portal Version 1.9.6.

Figure 4-3 Sample list of available fixpacks

3. Click the fixpack name and select the download option for the readme file.
4. Follow the instructions in the readme file to complete fixpack installation

If you know the fixpack number, you can retrieve the fixpack directly from the IBM FTP site. The Tivoli Enterprise Monitoring Server that is supplied with IBM Tivoli OMEGAMON XE V3.1.0 is at level 360 and the Tivoli Enterprise Portal Server is at level 196. These relate to the following URLs:

ftp://ftp.software.ibm.com/software/tivoli_support/patches/patches_3.6.0

or

ftp://ftp.software.ibm.com/software/tivoli_support/patches/patches_1.9.6

Figure 4-4 on page 119 shows the current FTP page for Tivoli Enterprise Portal Server.

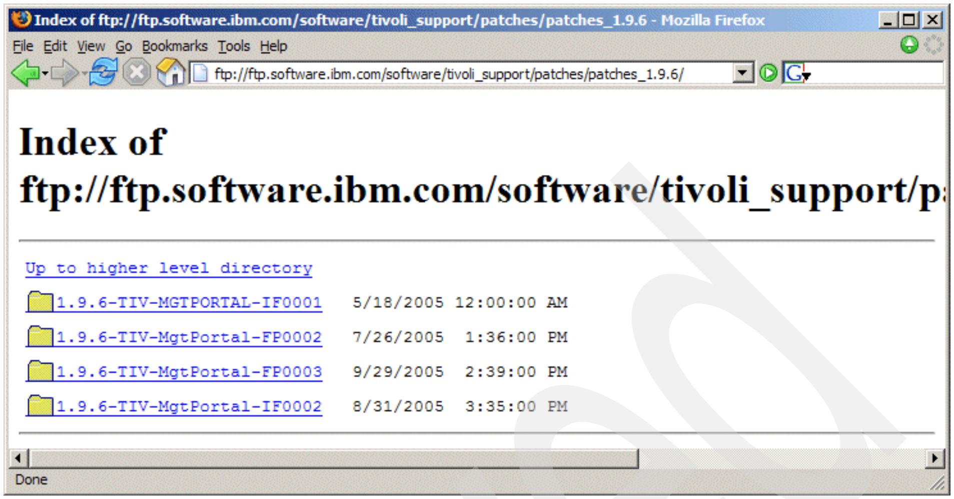


Figure 4-4 Tivoli Enterprise Portal Server fixpack list

4.3 OMEGAMON initialization checklist

This section provides a checklist for review in case of initialization problems during the startup of the OMEGAMON Tivoli Enterprise Monitoring Server address space. This checklist should be used in conjunction with IBM support, to help determine any configuration issues.

- 4.3.1, “RAS1 service initialization” on page 119
- 4.3.2, “ITMS:Engine service” on page 120
- 4.3.3, “Communication protocols” on page 121
- 4.3.4, “Connectivity to the hub” on page 123

4.3.1 RAS1 service initialization

The RAS acronym means Reliability, Availability, and Serviceability. The RAS1 service in this discussion refers to the basic services component used for diagnostic tracing. Nearly all diagnostic information for IBM Tivoli OMEGAMON XE is delivered through the RAS1 trace component. This component is configured in member KxxENV of RKANPAR using the KBB_RAS1 environment variable.

All OMEGAMON address spaces provide a RAS1 banner during RAS1 component initialization. Figure 4-5 on page 120 shows the RAS1 banner from our Tivoli Enterprise Monitoring Server in SC69.

```
=======================>  Candle CT/RAS1 Service Log +====================
     System Name: SC69                       Process ID: 93
       User Name: STC                          Job Name: CANSDSST
     System Type: OS/390;SP7.0.6         MAC1_ENV Macro: 0x1995
      Start Date: 2005/11/16                 Start Time: 13: 9:13
        KBB_RAS1: ERROR
     KBB_ENVPATH: KBBENV
==========================================================================
```

Figure 4-5 RAS1 banner

Successful initialization of RAS1 service with the presence of the RAS1 banner in Figure 4-5 is a necessary but not sufficient condition to diagnose failures in OMEGAMON address space. You also need to have the configuration of KBB_RAS1 to any value that is not NONE. In Figure 4-5, KBB_RAS1 is set to ERROR. This shows that all messages with the severity of error and above will be written out. This provides the necessary and sufficient level of RAS1 tracing to diagnose the majority of OMEGAMON address space startup issues.

You might also include the KDC_DEBUG=N parameter in the configuration.

4.3.2 ITMS:Engine service

The ITMS:Engine is a collection of basic z/OS and communication service routines built specifically for the z/OS environment. Most OMEGAMON address spaces load and employ the services of ITMS:Engine, except OMEGAMON classic components.

The ITMS:Engine successful initialization is noted by the KLVIN408 message as shown in Example 4-1.

Example 4-1 ITMS:Engine initialization

```
KLVIN408 CANDLE ENGINE VERSION 180 READY ON SC52(0A6A3A208
```

There are two possible ITMS:Engine initialization failures:

- Invalid ITMS:Engine startup parameters. Indicated by user abend code U0012.

  For U0012 abends, the startup parameter is referred by the RKLVIN DD statement of the started task JCL. Most often, U0012 abend can be remedied by backing out the last change made to the ITMS:Engine startup parameters. You also should check whether all STEPLIB datasets are APF authorized.

- Protocol initialization failures. Indicated by user abend code U0200.

  For U0200 abends, the root cause of the protocol failures must be remedied, as explained in the next section.

4.3.3 Communication protocols

This section discusses communication protocol initialization for TCP/IP and SNA service, and introduces the server list.

TCP/IP service initialization

The TCP/IP service provides end-to-end connectivity for application-layer codes, such as connectivity to Tivoli Enterprise Portal Server or a non-z/OS Tivoli Enterprise Monitoring Server.

> **Note:** If any TCP/IP bind messages are missing, you may have to change the RKANPAR(KDSENV) member parameter KDC_DEBUG to `Y` to get additional messages.

TCP/IP services for the address space are available if either of these messages is present in the RAS log:

```
"KDE1I_OpenTransportProvider") Transport opened: socket/ip.udp
"KDE1I_OpenTransportProvider") Transport opened: socket/ip.pipe
```

If neither message is in the log, refer to the RAS1 topic of this discussion to assure that the proper level of RAS1 tracing is in effect.

The first three messages in Example 4-2 are for IBM TCP/IP stack, and the last message indicates that the connection to Interlink TCP/IP failed, which is legitimate in our environment because it does not exist.

Example 4-2 RKLVLOG messages for TCP/IP

```
KLXIN001 TCP/IP CONFIGURATION: TCP/IP_USERID=TCPIP
KLXIN003 TCP/IP INTERFACE INITIALIZED
KLXIN001 HPNS INTERFACE AVAILABLE
...
...
KLXIN004 TCP/ACCESS INTERFACE INITIALIZATION BYPASSED
```

SNA service initialization

OMEGAMON address spaces must also have SNA support. SNA can be used exclusively or in conjunction with TCP/IP, as a transport service. This configuration is done during setup in configuration tool.

The messages in Example 4-3 are printed in the RAS1/RKLVLOG when the local SNA configuration is processed from the xxxENV member of RKANPAR.

Example 4-3 SNA initialization messages

```
"BSS1_GetEnv") KDCFP_ALIAS=KDCFC_ALIAS=KLXBS_ALIAS="K52DSLB"
"BSS1_GetEnv") KDCFP_TPNAME=KDCFC_TPNAME=KLXBS_TPNAME="SNASOCKETS"
"BSS1_GetEnv") KDCFP_MODE=KDCFC_MODE=KLXBS_MODE="CANCTDCS"
"getEnv") AF_SNA configuration: Alias(K52DSLB) Mode(CANCTDCS)
TpName(SNASOCKETS)
   .  .  .
"KDE1I_OpenTransportProvider") Transport opened: com1/sna.pipe
```

In Example 4-3:

`KDCFC_ALIAS` Identifies the APPL definition of the Independent Logical Unit to be used in this process.

`KDCFC_MODE` Identifies the LOGMODE name, the same name found in the LOGMODE specification of the KDCFC_Alias APPL definition.

`KDCFC_TPNAME` Transaction Processing Name.

The last message indicates that SNA is operational with the transport opened message.

Server list consideration

Many OMEGAMON processes build and query a list of possible Tivoli Enterprise Monitoring Server addresses, called the *server list*, which contains local location broker (LLB) and global location broker (GLB) entries. The LLB entries of the server list are derived, and the GLB entries are built from the content of the KDCSSITE member of RKANPAR. Example 4-4 shows a sample server list for our hub Tivoli Enterprise Monitoring Server.

Example 4-4 Server list for a hub Tivoli Enterprise Monitoring Server

```
"NewSDB") LLB entry 1 is ip.pipe:#9.12.4.42.1918.
"NewSDB") LLB entry 2 is ip:#9.12.4.42.1918.
"NewSDB") LLB entry 3 is sna:#USIBMSC.K52DSLB.CANCTDCS.SNASOCKETS.135.
"NewSDB") GLB entry 1 is ip.pipe:#9.12.4.42.1918.
"NewSDB") GLB entry 2 is ip:#9.12.4.42.1918.
"NewSDB") GLB entry 3 is sna:#USIBMSC.K52DSLB.CANCTDCS.SNASOCKETS.135.
"NewSDB") GLB entry 4 is ip.pipe:#9.12.4.42.1918.
"NewSDB") GLB entry 5 is ip:#9.12.4.42.1918.
"NewSDB") GLB entry 6 is sna:#USIBMSC.K52DSLB.CANCTDCS.SNASOCKETS.135.
KDSNC004   Bind of local location broker complete:
ip.pipe:#9.12.4.42.1918.
```

```
KDSNC004   Bind of local location broker complete: ip:#9.12.4.42.1918.
KDSNC004   Bind of local location broker complete:
sna:#USIBMSC.K52DSLB.CANCTDCS.SNASOCKETS.135.
KDSNC007   Local Location Broker is active
KDSNC005   Bind of global location broker complete:
ip.pipe:#9.12.4.42.1918.
KDSNC005   Bind of global location broker complete: ip:#9.12.4.42.1918.
KDSNC005   Bind of global location broker complete:
sna:#USIBMSC.K52DSLB.CANCTDCS.SNASOCKETS.135.
KDSNC008   Global Location Broker is active
```

Reviewing the server list is a task that support may consider when problem-solving server startup.

4.3.4 Connectivity to the hub

The last detail is the Tivoli Enterprise Monitoring Server authentication callback. Assuming that the application has received the address of the hub as shown in the server list messages and that the lbLookupHub() operation was successful, requests are sent to the hub.

At this point, the hub Tivoli Enterprise Monitoring Server has been contacted, but the application connection is not completed until the callback sequence completes. This set of messages illustrates the authentication callback sequence. The sequence in Example 4-5 must complete successfully.

Example 4-5 Handshake with the hub

```
"KDCL_GetBinding") Using LLB at ip.pipe:#9.12.6.66.1918.
"Mgr::locateEverbody") lbLookupHub returned error <0>,
   ip<ip:#9.12.6.66.1918.>
   sna<sna:#USIBMSC.K67DSLB.CANCTDCS.SNASOCKETS.135.>
   pipe <ip.pipe:#9.12.6.66.1918.>
"AtLexer::getKeyAndValue") <pkey> is not a valid Key!!!
"EntryNode::init") error: invalid entry <0>
"AtLexer::getKeyAndValue") <pkey> is not a valid Key!!!
"EntryNode::init") error: invalid entry <0>
"AtLexer::getKeyAndValue") <pkey> is not a valid Key!!!
"EntryNode::init") error: invalid entry <1>
"IBInterfaceCommon::registerIB") IB already registered for path
"IBInterfaceCommon::registerIB") Changing oldpath to new path
"IBInterface::readTableData") Fetch error is <0>
"KDCG_Bind") Using GLB at ip.pipe:#9.12.4.42.1918.
```

This last message in Example 4-5 on page 123 is key, as it indicates application session connectivity (or as commonly expressed, connected to the hub). In our hub Tivoli Enterprise Monitoring Server we see messages similar to Example 4-6. It shows two remote Tivoli Enterprise Monitoring Server connecting to our hub.

Example 4-6 Remote connecting to hub messages

```
KDSMA011   Logon successful to server SRVR01 user SITMON
ip.pipe:#9.12.6.66(1918).
KDSMA011   Logon successful to server SRVR01 user SRVR01
ip.pipe:#9.12.6.66(1918).
   .  .  .
KDSMA011   Logon successful to server SRVR01 user SITMON
ip.pipe:#9.12.6.22(1918).
KDSMA011   Logon successful to server SRVR01 user SRVR01
ip.pipe:#9.12.6.22(1918).
```

The messages in Example 4-7 indicate agent address spaces connecting to a Tivoli Enterprise Monitoring Server address space. The example shows a remote Tivoli Enterprise Monitoring Server.

Example 4-7 Agents connecting to Tivoli Enterprise Monitoring Server

```
"HeartbeatInserter") Remote node <SC67:CPIRA
"HeartbeatInserter") Remote node <WTSCPLX1:SC67:MVSSYS> is ON-LINE.
"HeartbeatInserter") Remote node <SC67:WTSC67:UNIX> is ON-LINE.
```

4.4 Important configuration members

To understand configuration of Tivoli Enterprise Monitoring Server on z/OS, the information is typically stored in RKANPAR and RKANCMD datasets. The information in these datasets may help resolve configuration and startup issues. The important configuration members are:

- Members in RKANPAR
  - KDSSYSIN
    - Pointers to all load lists KxxLLIST
    - Pointers to RKANCMD start members
    - Memory, LSR, Thread Parameters
  - KDSENV
    - IP parameters (also KDCSHOST and KDCSSITE)
    - KDS_RUN – Starts sysplex probes

– KxxLLIST – Module Load lists

– KxxINIT provides aliases for RKANPAR members

- Members in RKANCMD

  – KDSSTART: Startup command list for Tivoli Enterprise Monitoring Server

    - LBDAEMON: Local and global location brokers
    - DIALOG: Open ACBs for MQ and CUA Oper
    - IRAMAN: Agent in Tivoli Enterprise Monitoring Server

  – KxxAGST: Startup command list for agent-started tasks

    - IRAMAN: For agents in own address space

  – KMVSTKMV

    - IRAMAN for KWO and KMV, the SDM Bridge

4.5 Using the Service Console

With the Service Console enabled, you can get information about the running process, turn on and turn off traces without restarting the process, and manage environment variables. The Service Console is typically accessed from port 1920 using an HTTP request. Figure 4-6 shows an example of the page that lists the existing service consoles.

Figure 4-6 Candle Service Index

As shown in Figure 4-6 on page 125, each Tivoli Enterprise Monitoring Server and agent address space provides access to its own console. Going into the Tivoli Enterprise Monitoring Server console, as indicated by the started task name, opens the Service Console in Figure 4-7. This is an HTTP Service Console for a z/OS-based Tivoli Enterprise Monitoring Server address space.

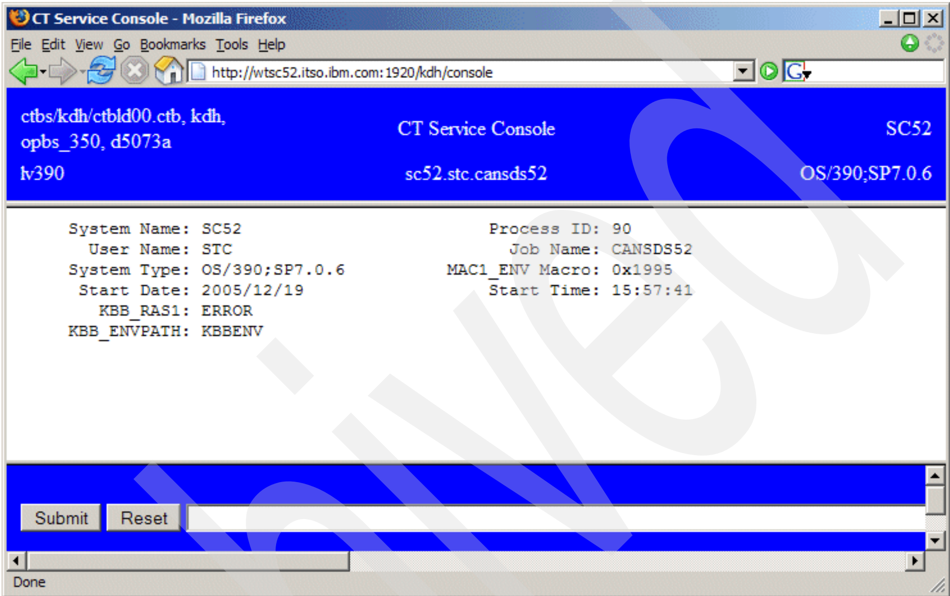


Figure 4-7 HTTP Service Console

You can list the commands that can be used from the Service Console by using the question mark, as shown in Figure 4-8.

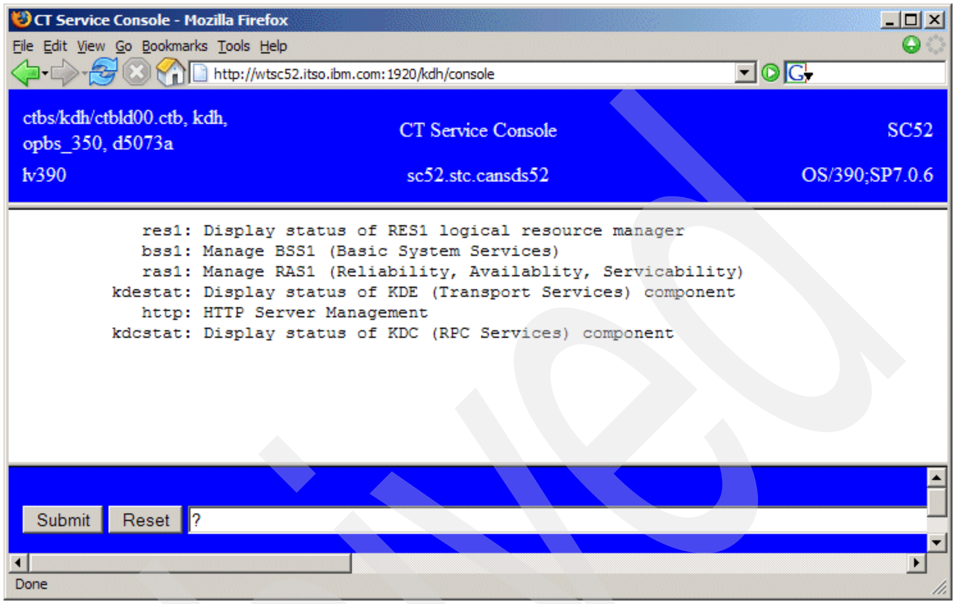


Figure 4-8 List of commands

In the context of problem determination, we focus on the **ras1** command. Figure 4-9 shows options for the **ras1** command.

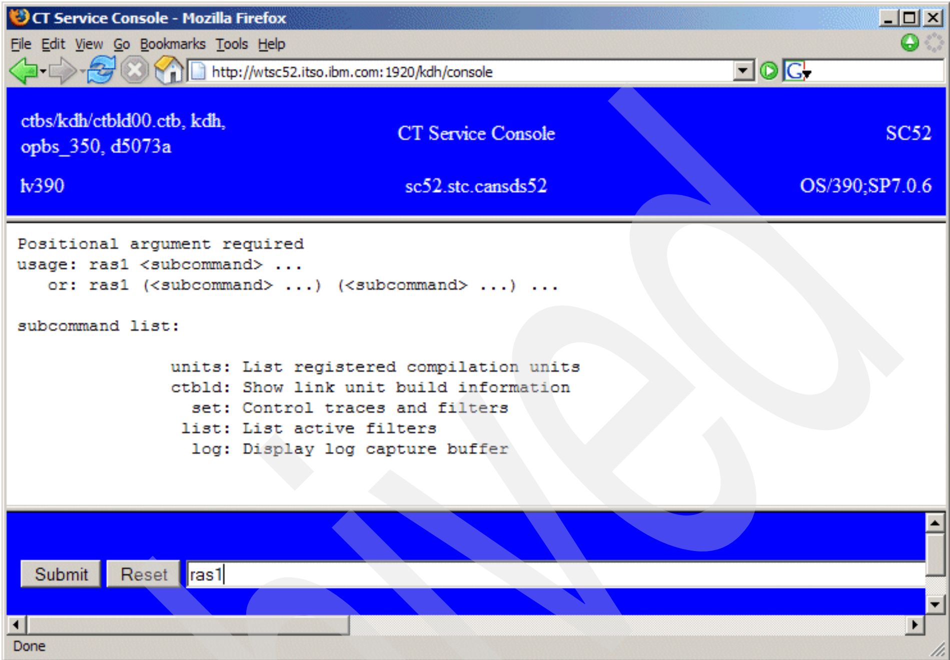


Figure 4-9 The ras1 command

The available subcommands for **ras1** are:

units Lists the registered compilation units that can have tracing activated or deactivated individually.

ctbld Lists the module build information that can be given to support.

set Activates and deactivates traces and filters.

list Shows the active filters.

log Retrieves the current log capture buffer, which is helpful for identifying an impending problem.

4.6 Logging and tracing

This section discusses various logging and tracing options of the IBM Tivoli OMEGAMON XE. The discussion includes:

- 4.6.1, “Setting trace information” on page 129
- 4.6.2, “Documentation program check” on page 130

- 4.6.3, “Unit tracing” on page 132
- 4.6.4, “Dynamic mainframe Tivoli Enterprise Monitoring Server traces” on page 132
- 4.6.5, “Dynamic agent traces” on page 133

4.6.1 Setting trace information

Occasionally, the IBM Support Center might ask you to set a particular trace option. You can use the **modify** command to enable additional tracing:

```
MODIFY [proc_name],KXDCMD SET DIAGLEV=n
```

n is between 1 and 9 inclusive. The trace levels above 4 are very verbose. You can turn off tracing by setting DIAGLEV to 1. Example 4-8 shows a sample RKLVLOG with changing the DIAGLEV setting to 4.

Example 4-8 Sample verbose log

```
2005.353 11:27:10.07 KLVOP191 REPLY FROM *MASTER*:
2005.353 11:27:10.07 KLVOP191    'KXDCMD SET DIAGLEV=4'
2005.353 11:27:10.07 EXDC COMMAND REQUESTED:  SET DIAGLEV=4
2005.353 11:27:10.07 TRACE_1624 -  - EXDC DIAGLEV HAS BEEN SET TO 4
2005.353 11:27:10.33 KXDSUB : SUBTASK WOKE ECB:40000004
2005.353 11:27:10.33 KXDSUB : THIS IS TOP OF CYCLE
2005.353 11:27:10.33 KXDSUB : RMF DATA AREA AT:216ECFE0 INTNUM:D83
2005.353 11:27:10.33 KXDSUBCP : INPUT PARMS:
2005.353 11:27:10.33 KXDSUBCP :              WORKA value   2541ACA8
2005.353 11:27:10.33 KXDSUBCP : LPAR CPU(7)
2005.353 11:27:10.33 KXDSUBCP : Cumm LPAR CPU(428)
2005.353 11:27:10.33 KXDSUBCP : MVS CPU(7)
2005.353 11:27:10.33 KXDSUBCP : Cumm MVS CPU(438)
2005.353 11:27:10.33 KXDSUB : START PROCESS CYCLE
2005.353 11:27:10.33 KXDCLPRC : CALL PROCESS FOR KXDWLCON AT
EP:23F31794
2005.353 11:27:10.33 KXDHSKP :  JOBNAME *MASTER* JOB START:
BE067B03D3BBB520 AS
2005.353 11:27:10.33 US: 41
2005.353 11:27:10.33 KXDHSKP :  JOBNAME PCAUTH JOB START:
BE067B03D3BBB520 ASCB
2005.353 11:27:10.33 : 41
2005.353 11:27:10.33 KXDHSKP :  JOBNAME RASP JOB START:
BE067B03D3BBB520 ASCBIN
2005.353 11:27:10.33 41
2005.353 11:27:10.33 KXDHSKP :  JOBNAME TRACE JOB START:
BE067B03D3BBB520 ASCBI
2005.353 11:27:10.33  41
```

```
2005.353 11:27:10.33 KXDHSKP :  JOBNAME DUMPSRV JOB START:
BE067B03D3BBB520 ASC
2005.353 11:27:10.33 S: 41
2005.353 11:27:10.33 KXDHSKP :  JOBNAME XCFAS JOB START:
BE067B03D3BBB520 ASCBI
2005.353 11:27:10.33  41
2005.353 11:27:10.33 KXDHSKP :  JOBNAME GRS JOB START: BE067B03D3BBB520
ASCBINT
2005.353 11:27:10.33 1
2005.353 11:27:10.33 KXDHSKP :  JOBNAME SMSPDSE JOB START:
BE067B03D3BBB520 ASC
2005.353 11:27:10.33 S: 41
   .  .  .
2005.353 11:27:10.33 KXDRQRY : RQRY RC =0
2005.353 11:27:10.34 KXDCLPRC : CALL PROCESS FOR KXDM3KCO AT
EP:23F084B0
2005.353 11:27:10.34 KXDM3KPR : KXDM3KPR - ENTERED
2005.353 11:27:10.34 KXDM3KPR : KXDM3KPR - PROCESS ROUTINE NORMAL EXIT
2005.353 11:27:10.34 KXDSUB : MAIN_DONE, EXDC_EVENTS:0
2005.353 11:27:10.34 KXDSUB : END OF CYCLE; ABOUT TO WAIT ON ECBS
2005.353 11:27:10.34 KXDSUB : SUBTASK WAIT ECB:4
```

You can force write the buffered messages and trace to RKLVLOG using this command:

```
MODIFY [proc_name],FLUSH
```

4.6.2 Documentation program check

When you see the message `OB0910 Program Check` when working with the OMEGAMON classic, you should obtain a screen print to send a report to IBM support. You should also issue the following OMEGAMON commands:

- The **`.PCS`** command collects information from the program check. When there is no program check, it returns the message `Program Check has not occurred`.
- The **`.ZAP`** command gives a dump of system levels as shown in Figure 4-10 on page 131.

```
________________          VTM     OM/DEX   V520./C SC52 02/02/06 15:00:58

.ZAP
+      Product Code:
+         Base = 01-000000-00000-00
+      Product = 01-000000-00000-00
+
+      MVS: SP7.0.6  CPU: 2084 #076A3A
+      MVS: SP7.0.6  CPU: 2084 #076A3A
+      MVS: SP7.0.6  CPU: 2084 #076A3A
+      Assembled with BASE V550, Executing with BASE V550
+      CANDLE Profile version V102 last modified on 10/11/05
+
+      Maintenance Applied by Component:
+       BASE     Version 550, level 00 - component identifier "OB"
+         PTF     - BASE OB550    BCM1 OB550    BMSG OB550    BR   OB550
+                   CMDV OB550    CSAR OB550    DELF OB550    DEV  OB550
+                   DIR  OB550    DSPC OB550    ENV  OB550    EXCD OB550
```

Figure 4-10   Sample ZAP command output

- The **.MOD** command lists the current modules and their levels as shown in Figure 4-11.

```
_______________          VTM     OM/DEX   V520./C SC52 02/02/06 15:02:39

.MOD
+      BASE     , Total Module Size = 381920
+|BASE 2254A0F0 3A40 V550 05/10/05 07:45|ISSS 22580930 1640 V550 05/10/05 07:47
+|BCM1 2254DB30 0900 V550 05/10/05 07:45|LEXC 22581F70 09E0 V550 05/10/05 07:47
+|BMSG 2254E530 4260 V550 05/10/05 07:45|LOGC 22582950 22A0 V550 05/10/05 07:47
+|BR   22552790 0080 V550 05/10/05 07:45|OMIO 22584BF0 1F40 V550 05/10/05 07:47
+|CMDV 22552810 2860 V550 05/10/05 07:46|PDEV 22586B30 07C0 V550 05/10/05 07:47
+|CSAR 22555070 04E0 V550 05/10/05 07:46|PEEK 225872F0 5960 V550 05/10/05 07:47
+|DELF 22555550 13C0 V550 05/10/05 07:46|PPRF 2258CC50 1040 V550 05/10/05 07:47
+|DEV  22556910 4960 V550 05/10/05 07:46|SHAR 2258DC90 22A0 V550 05/10/05 07:51
+|DIR  2255B270 31A0 V550 05/10/05 07:46|SRBD 2258FF30 0CC0 V550 05/10/05 07:52
+|DSPC 2255E410 27A0 V550 05/10/05 07:46|STUB 22590BF0 01E0 V550 05/10/05 07:52
+|ENV  22560BB0 0A40 V550 05/10/05 07:46|SUB  22590DD0 2500 V550 05/10/05 07:52
+|EXCD 225615F0 34A0 V550 05/10/05 07:46|SUBE 225932D0 0F00 V550 05/10/05 07:52
+|GDFN 22564A90 1240 V550 05/10/05 07:46|SUBX 225941D0 1CE0 V550 05/10/05 07:52
+|ICMD 22565CD0 5220 V550 05/10/05 07:46|SUB1 22595EB0 1D00 V550 05/10/05 07:52
+|ICM1 2256AEF0 0CE0 V550 05/10/05 07:46|SUB2 22597BB0 0E60 V550 05/10/05 07:52
```

Figure 4-11   Sample MOD command output

4.6.3 Unit tracing

If you do not have the Service Console enabled, you can request a RAS1 trace request by specifying unit trace in the KBB_RAS1 statement. For example, to turn on unit trace in RKANPAR(KxxENV) we use a statement similar to Example 4-9.

Example 4-9 RAS1 error trace

```
KBB_RAS1=ERROR(UNIT:KRAAFIRA STATE) (UNIT:KRAADSPT STATE)
```

The KRAAFIRA and KRAADSPT unit traces enable you to see the data collection messages in the RKLVLOG on z/OS or the distributed RAS1 log in CANDLE/CMS/LOGS/KxxRAS1.LOG. Example 4-10 shows sample data collection messages. In this example, the situation name could be _Z_ if it has been synchronized, a user situation REG SIT, a Tivoli Enterprise Portal query, or a historical situation, in which case the situation name would be blank. It also shows the number of situations sent.

Example 4-10 KFAAFIRA trace response

```
"KRAAFIRA.cpp, Drive Data collection" agenttype.tablename <req#>
sitname
(KRAADSPT.cpp,"send data to Proxy") Sending n rows for Sitname
agenttype.tablename <req#>
```

Similar results can be achieved using the Service Console. Example 4-11 shows the command to enable the unit trace for data collection.

Example 4-11 RAS1 trace specification

```
RAS1 SET 'ERROR (UNIT:KRAAFIRA STATE) (UNIT:KRAADSPT STATE)'
```

4.6.4 Dynamic mainframe Tivoli Enterprise Monitoring Server traces

The CTDS control command is only for the Tivoli Enterprise Monitoring Server address space and does not work for agent address spaces. To use the CTDS control command, create a member in RKANCMD with the desired trace commands, then issue a MODIFY command against the Tivoli Enterprise Monitoring Server and specify the member name. For example, to start trace using the TRCON member shown in Example 4-12, issue the command F CANSDSST,TRCON.

Example 4-12 RKANCMD member TRCON for trace activation

```
CTDS TRACE ADD FILTER ID=AFIRA UNIT=KRAAFIRA CLASS(STATE)
CTDS TRACE ADD FILTER ID=AROW UNIT=KRAADSPT CLASS(STATE)
```

Use the TRCOFF member shown in Example 4-13 to stop the trace by invoking the command **F CANSDSST,TRCOFF**.

Example 4-13 RKANCMD member TRCOFF for trace termination

```
CTDS TRACE REMOVE FILTER ID=AFIRA
CTDS TRACE REMOVE FILTER ID=AROW
```

Example 4-14 shows the sample member for starting trace for situation monitor.

Example 4-14 Tracing situation

```
CTDS TRACE ADD FILTER ID=ASYNC UNIT=KO4ASYNC CLASS(ERROR STATE FLOW)
CTDS TRACE ADD FILTER ID=SITMA UNIT=KO4SITMA CLASS(ALL)
CTDS TRACE ADD FILTER ID=TOBJE UNIT=KO4TOBJE CLASS(ALL)
```

4.6.5 Dynamic agent traces

The agent traces can also be dynamically started from the Tivoli Enterprise Portal Server. From the managed system list, select **Take Action**, then enter the system command as shown in Figure 4-12. Use the argument:

```
TRACE: ADD FILTER ID=AFIRA UNIT=KRAAFIRA CLASS(STATE)
```

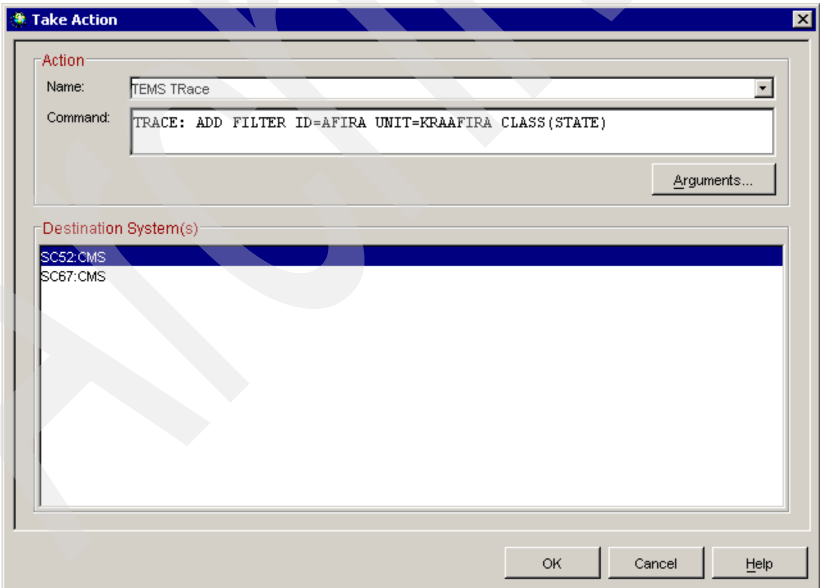


Figure 4-12 Take Action for Tivoli Enterprise Monitoring Server trace

5

IBM Tivoli OMEGAMON XE performance optimization

This chapter discusses OMEGAMON performance and some considerations for distributed connection. The chapter includes the following topics:

5.1 General performance guidelines

This section provides a general overview of some performance issues and guidelines for IBM Tivoli OMEGAMON XE itself. The discussion consists of the following topics:

- 5.1.1, “IBM Tivoli OMEGAMON XE performance” on page 136
- 5.1.2, “IBM Tivoli OMEGAMON XE processing overview” on page 137
- 5.1.3, “General performance guidelines” on page 141

5.1.1 IBM Tivoli OMEGAMON XE performance

Figure 5-1 shows the architecture of IBM Tivoli OMEGAMON XE.

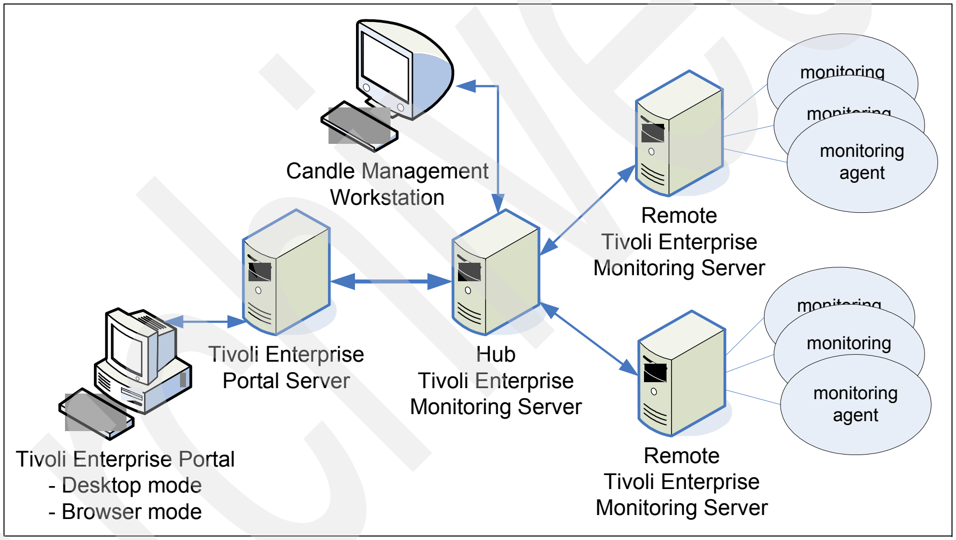


Figure 5-1 IBM Tivoli OMEGAMON XE architecture

As discussed in Chapter 2, “IBM Tivoli OMEGAMON XE components and architecture” on page 19, this architecture allows a scalable implementation of monitoring infrastructure. The architecture scalability supports the following topology when using the IBM Tivoli Monitoring V6.1 infrastructure. These figures are based on internal benchmarking. Your topology may vary—either larger or smaller—based your workload and your combination of agents, situations, policies, amount of data, and so on.

- Overall the hub Tivoli Enterprise Monitoring Server can manage up to 4000 managed systems. This number consists of nodes and sub-nodes. A monitoring agent is considered a node, but on z/OS, separate subsystems that are managed by a single agent can be considered sub-nodes (such as a CICS address space or a WebSphere MQ queue manager).

- A stand-alone or remote Tivoli Enterprise Monitoring Server can manage up to 500 managed systems. This is assuming a default heartbeat interval for all nodes. For historical data collection, a limit of 250 agents is supported.
- A hub Tivoli Enterprise Monitoring Server supports at least 15 remote Tivoli Enterprise Monitoring Servers. You cannot have all remote Tivoli Enterprise Monitoring Servers running at full capacity of 500 managed agents, which violates the limit of 4000 managed systems.
- Supports 50 simultaneous Tivoli Enterprise Portal clients connected to Tivoli Enterprise Portal Server, of which 50 clients actively retrieve real-time data.

To be able to optimize the scale listed above, the environment must be optimized in its performance, the performance of IBM Tivoli OMEGAMON XE itself. This chapter covers various considerations related to IBM Tivoli OMEGAMON XE performance.

5.1.2 IBM Tivoli OMEGAMON XE processing overview

In general, IBM Tivoli OMEGAMON XE processing involves the following steps:

- Users connect using Tivoli Enterprise Portal, open a workspace, retrieve data from one or more queries, and refresh the display periodically.
- Event situations fire at pre-defined intervals to evaluate system health. Some of these situations are executed by the situation monitor and some are invoked from policies (workflow automation).
- Background data collectors (such as UADVISORs) retrieve monitoring data and store the data in virtual tables (VTABLEs).
- Optional historical interval data collection is conducted by Tivoli Enterprise Monitoring Agents.

Based on these high-level processing functions of IBM Tivoli OMEGAMON XE, we can view trace data and break down the processing requirements for each process and determine how to improve the overall performance.

User workspace data retrieval

On-demand overhead occurs when the user requests data. Every time you open a workspace, the Tivoli Enterprise Portal Server requests a data sample as specified in each workspace view’s query. Every time that workspace is refreshed, automatically at 30-second to 60-minute intervals or when you click Refresh, the server requests another sample.

Workspaces, or more accurately workspace views, with many data rows being returned and workspaces with automatic refresh can significantly increase processing. You can use custom queries to potentially reduce the processing of auto-updating workspaces.

The operator data retrieval process can be summarized in Figure 5-2.

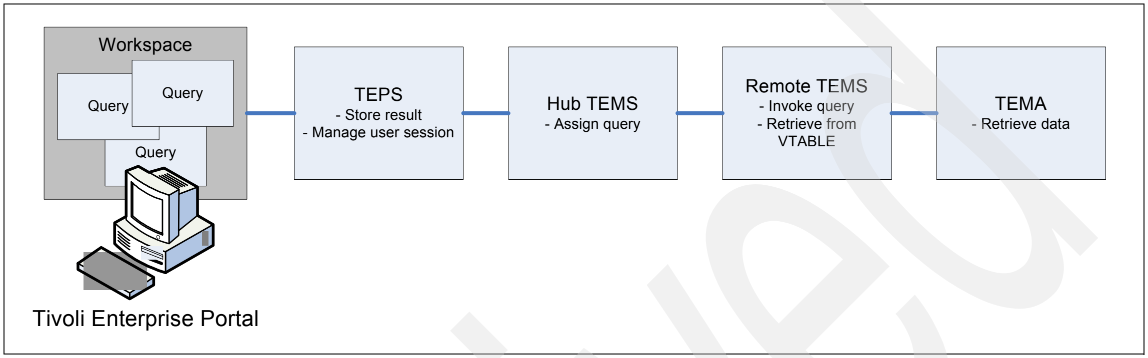


Figure 5-2 Operator data retrieval

As shown in Figure 5-2, the processing goes through:

- Tivoli Enterprise Portal requires processing to show and store data for each page of information and the last page that it opened. The more queries there are for a workspace, the more memory is allocated to back it up. As the Tivoli Enterprise Portal is a Java application, the main bottleneck is typically the memory. Another potential bottleneck is that data and workspace information are constantly retrieved from Tivoli Enterprise Portal Server, therefore good network performance between the client and the server is essential.
- Tivoli Enterprise Portal Server stores each result being viewed from the client so it does not have to pull the data again from Tivoli Enterprise Monitoring Server unless the user refreshes the workspace. There are typically multiple queries per workspace. It may be useful to have a workspace that presents different information from the same query, as this will reduce the processing requirement from the Tivoli Enterprise Portal Server upwards. As Tivoli Enterprise Portal Server is a Java-based server, the main concern is for memory.
- The hub Tivoli Enterprise Monitoring Server manages all requests from Tivoli Enterprise Portal Server and assigns the queries to the appropriate remote Tivoli Enterprise Monitoring Server. If the Tivoli Enterprise Monitoring Agent is connected directly to the hub Tivoli Enterprise Monitoring Server, then Tivoli Enterprise Monitoring Server invokes the query.

- Remote Tivoli Enterprise Monitoring Servers invoke the query on the agent or, in the case of a background data collection (such as UADVISORs), it retrieves the data from the in-storage virtual tables.
- The agent collects the information for the query and returns the metrics to the Tivoli Enterprise Monitoring Server.

Situation data retrieval

Situations perform data collection. There are two types of situations:

- Situations that trigger data collection at the agent. You may be able to synchronize their intervals to make one data sample do the work of several situations. See 5.4.5, “Situation synchronization” on page 168.
- Situations that work with background data collection. Typically the background collection is driven by UADVISOR collection to VTABLEs. The background collection is used in some z/OS-based solutions such as IBM Tivoli OMEGAMON XE for z/OS and IBM Tivoli OMEGAMON XE for WebSphere Business Integration on z/OS.

Although UADVISORs are not exposed in the user interface, you can see evidence of them in log messages, such as the RKLVLOG in Example 5-1.

Example 5-1 UADVISOR in RKLVLOG

```
KO41036    Monitoring for situation UADVISOR_KM5_VCF_PATHS started.
KO41036    Monitoring for situation UADVISOR_KM5_VWSWMD started.
KO41036    Monitoring for situation UADVISOR_KM5_VXCF_PATH started.
KO41036    Monitoring for situation UADVISOR_KM5_VXCF_SSTAT started.
```

Situations are processed by situation monitor or SITMON process. This can happen at various levels. The hub Tivoli Enterprise Monitoring Server initiates situation processing.

- Tivoli Enterprise Monitoring Agent retrieves and analyzes the data with local SITMON. If the situation becomes true, all true rows are returned; otherwise it will not return a row. Tivoli Enterprise Monitoring Agent informs Tivoli Enterprise Monitoring Server of any situation state change.
- When a situation contains column functions, all rows are returned to remote Tivoli Enterprise Monitoring Server which evaluates in its SITMON.
- When a situation contains an embedded situation from a Tivoli Enterprise Monitoring Agent that is connected to a different Tivoli Enterprise Monitoring Server, it will be evaluated in the hub Tivoli Enterprise Monitoring Server.
- Hub is informed of all situation state changes and stores them in a situation history file and forwards these events to Tivoli Enterprise Portal Server.

Figure 5-3 shows the situation processing.

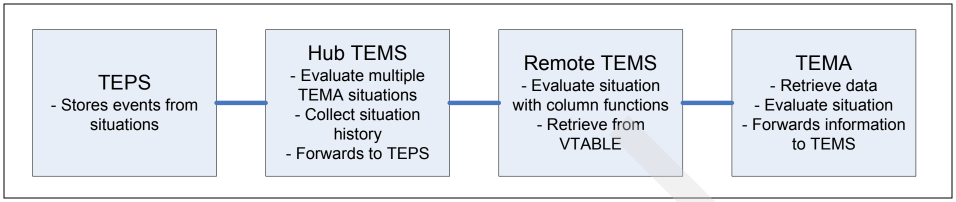


Figure 5-3 Situation data retrieval

Historical data collection

Historical data collection is configured from Tivoli Enterprise Portal client and can be set for 5-minute, 15-minute, 30-minute, or one-hour intervals. Most IBM Tivoli OMEGAMON XE products on z/OS collect historical data with UADVISOR background data collection. If the historical metrics are to be offloaded into the data warehouse, the warehousing interval can be either hourly (once every hour) or daily (once every 24 hours).

The warehouse proxy connects to either the agent directly or the remote Tivoli Enterprise Monitoring Server depending on where the short-term historical metrics are stored, as shown in Figure 5-4.

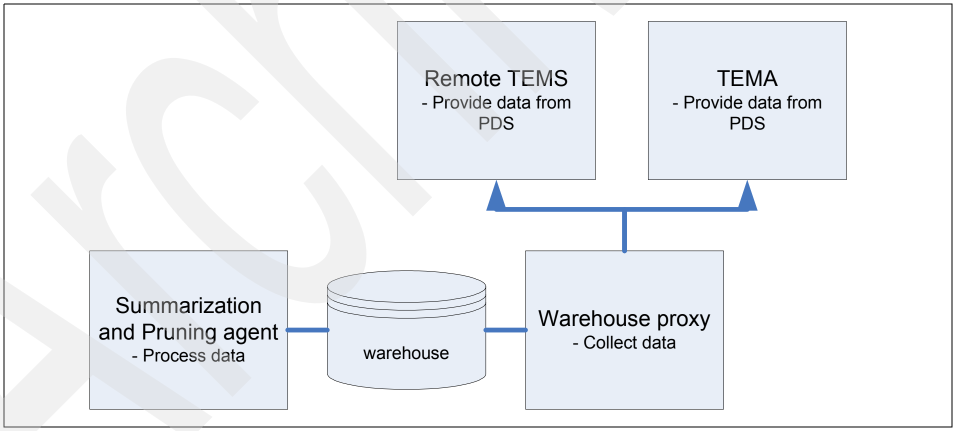


Figure 5-4 Historical data processing

Data is collected from the Persistent Data Store (PDS) residing in Tivoli Enterprise Monitoring Server or agent by the warehouse proxy to be stored in the historical data warehouse. The summarization and pruning agent processes the collected data in the warehouse, creates summary information, and optionally prunes the detail data from the database.

Note: In IBM Tivoli OMEGAMON XE V3.1 there is no support for the summarization and pruning function for data coming from z/OS. Summarization and pruning agent is available for IBM Tivoli Monitoring V6.1.

To minimize bulk data transfer, it is highly recommended that the warehouse proxy and summarization and pruning agent run on the same machine as the warehouse database.

5.1.3 General performance guidelines

Based on the processing overview discussed in 5.1.1, “IBM Tivoli OMEGAMON XE performance” on page 136, we suggest some generic performance guidelines that are based on collecting performance information from IBM Tivoli OMEGAMON XE running on the z/OS platform. These general guidelines are:

- Use one or more remote Tivoli Enterprise Monitoring Servers to connect agents in a large environment. The benefits of using this structure are:
  - You can off-load agent management processing from the hub Tivoli Enterprise Monitoring Server so it can perform hub-related work.
  - Provides support for more Tivoli Enterprise Monitoring Agents relative to using a single Tivoli Enterprise Monitoring Server.
  - Provides a heartbeat concentrator to minimize network traffic for hub Tivoli Enterprise Monitoring Server. The remote Tivoli Enterprise Monitoring Server sends a heartbeat summary to the hub Tivoli Enterprise Monitoring Server.
  - Shortens the distance traveled for agent heartbeat and other agent-related processing. For example, a geographically local remote Tivoli Enterprise Monitoring Server is more responsive to agent processing. This topology can be useful for geographically dispersed enterprises. Also, policies run on these remote Tivoli Enterprise Monitoring Servers are more responsive and hub tolerant.
- Minimize ASCII-to-EBCDIC conversions. Whenever data passes between z/OS and distributed machines, the *receiver* converts the data to ASCII or EBCDIC as required. These conversions can happen for the following cases:
  - Distributed agents connect to a z/OS-based Tivoli Enterprise Monitoring Server.
  - z/OS-based Remote Tivoli Enterprise Monitoring Servers connect to a distributed-based hub Tivoli Enterprise Monitoring Server.
  - A z/OS-based hub Tivoli Enterprise Monitoring Server connects to Tivoli Enterprise Portal Server.

– Warehouse proxy connects to a z/OS-based agent or z/OS-based Tivoli Enterprise Monitoring Server.

In all cases, it is advisable to limit the data conversion for all required processing. As the warehouse proxy and Tivoli Enterprise Portal Server are available only on ASCII platforms (Windows, UNIX, Linux, even Linux on zSeries), z/OS-based data must go through conversion at least once.

Examples of redundant data conversion that can be avoided by having a distributed-based Tivoli Enterprise Monitoring Server include:

– Distributed agent connected to z/OS-basedTivoli Enterprise Monitoring Server and retrieved byTivoli Enterprise Portal Server

– Warehouse proxy connected to z/OS-based Tivoli Enterprise Monitoring Server for retrieving data by Tivoli Enterprise Portal

- Using individual address space adds little to CPU and memory requirements and can improve problem determination by compartmentalizing the problem. A typical recommendation is to run Tivoli Enterprise Monitoring Agents in their own address spaces if possible. A single Tivoli Enterprise Monitoring Agent per address space provides the following benefits:

  – Reduces problem source identification time

    You can more easily isolate and identify the source of problems when the address space contains only one solution.

  – Continuous operation

    With multiple address spaces, you can make a small change or upgrade the agent and cycle one address space instead of quiescing all the others. If you have a problem that is causing your agent address space to terminate, you lose one component instead of multiple.

  – Startup time

    Startup time for each address space is significantly reduced. When a Tivoli Enterprise Monitoring Server or Tivoli Enterprise Monitoring Agent starts (or re-starts), many services are engaged, one of which is the historical persistent data store. These services all have to initialize before startup is complete. If you have the PDS datasets in a different address spaces, it will not affect Tivoli Enterprise Monitoring Server startup time. If you have multiple-engine hardware, they can happen concurrently. Many of the address spaces preload all of their code. If you have multiple products, it has to load multiple products; if you have one product, it only loads that one plus the ITMS:Engine (previously referenced as the CT/Engine).

- Minimize data retrieval on the agent, which relates to optimizing the situations and queries. Situations are driven at intervals. In CT350 and CT360, policies (workflow automation) drive their own situations to take their own samples. In

IBM Tivoli Monitoring Services V6.1, policies will not drive an additional sample. Queries drives data collection based on user requests.

- Remove duplicate data collection and redundant address spaces. Processing overhead such as OMEGAMON II CUA address spaces do not have to be active for most of the data that is collected by IBM Tivoli OMEGAMON XE.

Based on these general performance guidelines, we discuss specific performance analysis in the rest of this chapter. Analyses are presented for the following topics:

- Tivoli Enterprise Monitoring Server considerations affecting the processing of data to and from the agent and portal. This is discussed in 5.2, “Tivoli Enterprise Monitoring Server performance” on page 143.
- Tivoli Enterprise Portal Server, Tivoli Enterprise Monitoring Server, and Tivoli Enterprise Monitoring Agent query processing is discussed in 5.3, “Query optimization and tuning” on page 158.
- Situations, another major item that must be well-tuned as these can be a source of processing even without any active users on the system, in 5.4, “Tuning situations” on page 162.
- Product-specific performance optimization in 5.5, “Product-specific tuning” on page 175.

5.2 Tivoli Enterprise Monitoring Server performance

The Tivoli Enterprise Monitoring Server, as the main processing component for IBM Tivoli OMEGAMON XE products, plays a vital part for the solutions. The hub Tivoli Enterprise Monitoring Server acts as the central consolidation point for all monitoring environments in the system. Typically the Tivoli Enterprise Monitoring Server is a relatively more CPU-intensive system. It is not necessarily memory intensive nor i/o intensive, except for the optional persistent data stores (PDSs). This section is dedicated to analyzing the CPU behavior of Tivoli Enterprise Monitoring Server:

- 5.2.1, “Major processes in Tivoli Enterprise Monitoring Server” on page 144
- 5.2.2, “Tivoli Enterprise Monitoring Server hub platforms” on page 146
- 5.2.3, “Tivoli Enterprise Monitoring Server overhead isolation procedures” on page 148
- 5.2.4, “Disabling products and functions” on page 155

5.2.1 Major processes in Tivoli Enterprise Monitoring Server

The processing requirement in Tivoli Enterprise Monitoring Server that typically requires CPU power can be listed as follows:

- Situation evaluation
- Situation state changes
- Managing Tivoli Enterprise Monitoring Agents
- Minimizing ASCII / EBCDIC conversions
- Historical data ASCII-to-EBCDIC conversion
- Historical data storage in monitoring agents
- Data movement and sorting
- Tivoli Enterprise Monitoring Server resident data collection
- Historical for resident collection

Situation evaluation

The Tivoli Enterprise Monitoring Agent evaluates most situations, but some predicate evaluations must be performed on Tivoli Enterprise Monitoring Server, such as column operation. Column operations, such as counting of occurrences or determining a maximum or minimum value, cannot be performed at the agent. Tivoli Enterprise Monitoring Server retrieves data from the agent and performs the column function. This means higher processing requirements at the Tivoli Enterprise Monitoring Server. You should try to prevent this processing occurring at the hub Tivoli Enterprise Monitoring Server, if possible.

Situation state changes

Typically, situation state changes are a small part of hub Tivoli Enterprise Monitoring Server processing. If not, it is usually easy to identify and tune. Every time a situation changes state, the following processing occurs in the Tivoli Enterprise Monitoring Server:

1. The Tivoli Enterprise Monitoring Agent informs its local Tivoli Enterprise Monitoring Server regarding the state change and may start a policy thread, process imbedded situations, or both.
2. The state change is forwarded to the hub Tivoli Enterprise Monitoring Server where it is logged to DASD; if it was an open event, the entire row of data that evaluated true is written to the history log.

False-positive events where the evaluation criteria are invalid for the environment (such as threshold defined inaccurately), especially in high volume, are unnecessary processing. Review the 24-hour situation raised history bar chart on the default Enterprise navigator item and evaluate any situation that shows up more than 100 times.

Managing Tivoli Enterprise Monitoring Agents

When agents are connected to the hub Tivoli Enterprise Monitoring Server, it has to process their heartbeats. With a large number of agents, this can consume measurable CPU and slow the responsiveness of the clients. The overhead can be offloaded to a remote Tivoli Enterprise Monitoring Server. A remote Tivoli Enterprise Monitoring Server also assists in situation distribution through a fan-out process, in which the hub Tivoli Enterprise Monitoring Server distributes one situation to the remote Tivoli Enterprise Monitoring Server and the remote Tivoli Enterprise Monitoring Server distributes that situation to every one of its agents on the distribution list.

Minimizing ASCII / EBCDIC conversions

Whenever data passes between z/OS and distributed machines (Linux on zSeries also is based on ASCII), the *receiver* converts the data to ASCII or EBCDIC as required. Where distributed agents connect to a z/OS-based Tivoli Enterprise Monitoring Server, the Tivoli Enterprise Monitoring Server must convert ASCII data to EBCDIC for every request. To eliminate the conversion, connect distributed agents to a distributed hub or remote Tivoli Enterprise Monitoring Server. The configuration that causes the highest conversion processing is a distributed Tivoli Enterprise Monitoring Agent connected to a z/OS-based hub Tivoli Enterprise Monitoring Server connected to Tivoli Enterprise Portal Server. Both the hub Tivoli Enterprise Monitoring Server and the Tivoli Enterprise Portal Server must convert the data.

Remote Tivoli Enterprise Monitoring Servers enable you to offload data conversion processing from the hub Tivoli Enterprise Monitoring Server. For example, if your hub Tivoli Enterprise Monitoring Server is hosted on z/OS, by attaching distributed Tivoli Enterprise Monitoring Agents to a z/OS remote Tivoli Enterprise Monitoring Server, the ASCII-to-EBCDIC conversion will occur in the remotes. Because the Tivoli Enterprise Portal Server is always on an ASCII platform (such as Windows, UNIX, Linux, or Linux on zSeries) there will always be at least one EBCDIC-to-ASCII conversion.

Historical data ASCII-to-EBCDIC conversion

If you store historical data at the agent as recommended, there is no Tivoli Enterprise Monitoring Server data conversion. If you collect history at the Tivoli Enterprise Monitoring Server where agents of different protocols are connected and it's being converted to EBCDIC, the data must be converted again to ASCII when offloaded to a data warehouse. The conversions are reversed if the data is retrieved from the warehouse: distributed warehouse to z/OS-based Tivoli Enterprise Monitoring Server to distributed based Tivoli Enterprise Portal Server.

Historical data storage in monitoring agents

Avoid historical data storage at the hub Tivoli Enterprise Monitoring Server if possible. Instead, store historical data at the agent or remote Tivoli Enterprise Monitoring Server. If you store history in the PDS (Persistent Data Store) files, it may be converted as above but it must be written to DASD. This can use cycles to store and retrieve. Besides steady state cycles, large or multiple PDSs affect Tivoli Enterprise Monitoring Server restart time, as they are read completely at startup.

> **Note:** Two PTFs exist, UA19870 (for Candle Management Server V350) and UA19871 (for Candle Management Server V360), that significantly reduce the processing requirements when accessing Persistent Data Store (PDS) datasets.

5.2.2 Tivoli Enterprise Monitoring Server hub platforms

When we move the hub Tivoli Enterprise Monitoring Server from z/OS, some of the processing load is offloaded from the z/OS-based Tivoli Enterprise Monitoring Server and executes on the distributed platform. This can help to reduce the processing need for mission-critical z/OS systems to a dedicated server.

The following processing is offloaded from the z/OS-based Tivoli Enterprise Monitoring Server:

- Situation state changes
- Situation evaluation
- Managing agents (heartbeats and distribution)
- Data conversions: ASCII-to-EBCDIC or EBCDIC-to-ASCII
- Historical data storage from remote agents

However, often the largest source of hub Tivoli Enterprise Monitoring Server processing is code that must run on z/OS. But this code does not necessarily have to run in the hub. Some of this overhead can be moved to other z/OS images, while the remainder can stay on the same machine, even if the hub is moved to a distributed platform. You can experience higher CPU in the hub Tivoli Enterprise Monitoring Server than in other z/OS-based Tivoli Enterprise Monitoring Servers. This may occur due to:

- Data movement and sorting

  Moving large quantities of data can be a major source of processing, especially if the data is sorted or merged. This overhead can usually be moved to a remote Tivoli Enterprise Monitoring Server or tuned.

- Tivoli Enterprise Monitoring Server resident data collectors
  - OMEGAMON II for MVS component's WLM probe

    The OMEGAMON II for MVS component of IBM Tivoli OMEGAMON XE for z/OS WLM probe must remain on an LPAR in a Tivoli Enterprise Monitoring Server, which can be a hub or a remote. If OMEGAMON II for MVS CUA users have auto-update turned on, the number of users and the auto-update interval may cause more processing on one LPAR, such as the hub, than another.
  - Hub Tivoli Enterprise Monitoring Server probes

    Any products executing within a Tivoli Enterprise Monitoring Server such as IBM Tivoli OMEGAMON XE for z/OS cannot be moved. You can still have a remote Tivoli Enterprise Monitoring Server on the z/OS image. Higher CPU usage by these functions on the hub Tivoli Enterprise Monitoring Server may be related to the number of subsystems or the volume of their workload.
  - Sysplex proxy

    If you have IBM Tivoli OMEGAMON XE for z/OS, it is likely that the sysplex proxy data consolidation function has defaulted to running in the hub Tivoli Enterprise Monitoring Server. This is a high-processing requirement function and can be relocated easily. Simply configure the hub Tivoli Enterprise Monitoring Server as not eligible for the sysplex proxy, which will offload cycles from the hub Tivoli Enterprise Monitoring Server. This is typically not done because the first Tivoli Enterprise Monitoring Server defined is the hub Tivoli Enterprise Monitoring Server and, by default, the configuration tool assigns the first Tivoli Enterprise Monitoring Server as the sysplex proxy. You need to change this using IBM Configuration Tool after the first remote Tivoli Enterprise Monitoring Server has been defined to the hub. As Figure 5-5 on page 148 indicates, the hub Tivoli Enterprise Monitoring Server has been excluded from performing the sysplex proxy function.

```
--------------------- List of RTEs assigned to a Sysplex ---------------------
COMMAND ===>

 Current Sysplex:     *WTSCPLX1
 * Use WTSCPLX1 Sysplex name when building managed system name ==> Y (Y, N)
 Sysplex Desc:         ITSO Plex
 Current ENQPlex:      ITSOENQ
 ENQPlex Desc:         ITSO Enqueue Plex
 Datastore high-level qualifier: OMEGAMON.WTSCPLX1

RTE       RTE                                         Proxy      CMS
name      description                                 position   type
SHR1      Sharing RTE for SC52                        EXCLUDE    HUB
SHR2      Sharing RTE for SC69                        BACKUP     REMOTE
SHR3      Sharing RTE for SC67                        BACKUP     REMOTE
 F1=Help  F3=Back  F7=Up  F8=Down
```

Figure 5-5 Tivoli Enterprise Monitoring Server sysplex proxy eligibility

> **Note:** A major component of the sysplex proxy processing is often the consolidation of DASD data for the Shared DASD report. This data can be reduced significantly by using a DASD Threshold Filtering situation. See "Shared DASD probe" on page 184.

- Historical data from resident collectors

5.2.3 Tivoli Enterprise Monitoring Server overhead isolation procedures

As discussed before, the primary processing in the Tivoli Enterprise Monitoring Server is either user requests or situation firing. Not much can be done for ad hoc user requests. Some user-related tuning is discussed in 5.3, "Query optimization and tuning" on page 158, which mainly evaluates processing related to situations.

Typically, situation processing isolation must be done in conjunction with the situation tuning discussed in 5.4, "Tuning situations" on page 162. Usually you want to autostart all of your required situations to optimize processing with situation synchronization, then isolate the problem using either of the following methods:

- Running IBM Tivoli OMEGAMON XE for z/OS PSW sampler INSPECT for z/OS-based Tivoli Enterprise Monitoring Server enables you to identify the module that consumes CPU. Positive identification can be performed after disabling situations related to the modules and remeasuring the performance.

Note: INSPECT is available from the OMEGAMON II for MVS 3270 interface and the Tivoli Enterprise Portal. If INSPECT for Tivoli Enterprise Monitoring Server, use the OMEGAMON II interface; otherwise, you will add processing requirements to the Tivoli Enterprise Monitoring Server to display the metrics that in turn will add more data to the INSPECT results.

- Measuring CPU second usage of Tivoli Enterprise Monitoring Server and disabling one-by-one the product's situations. This allows identification on which the solution's situation contributes the largest amount of CPU.

Note that some of the situation overhead may also be generated by UADVISOR situations. These UADVISOR situations are collected on their own interval.

The following sections provide detailed procedures for problem isolation.

Isolating a CPU consumption source using INSPECT

The following procedure can be used to investigate Tivoli Enterprise Monitoring Server processing using the OMEGAMON II for MVS component of the IBM Tivoli OMEGAMON for z/OS INSPECT facility. This process is similar to the process you would use to use INSPECT to investigate any address space. This is a high-level overview of the procedure.

Note: If you need to investigate Tivoli Enterprise Monitoring Server address space CPU consumption at this level of detail, it should be done with assistance from an IBM Tivoli IT Specialist, an IBM Software Services for Tivoli (ISST) Consultant, or an IBM Software Support Specialist. Additionally, a Problem Management Record (PMR) should be opened so the issue can be properly tracked and any defects properly documented.

1. With all situations running, use the OMEGAMON II for MVS component's INSPECT function and allow it to run for about six minutes. This is one minute longer than the highest situation interval (including the UADVISOR intervals), so it ensures that all background data collection executions are captured. See also 5.4.4, "Situation threshold and interval" on page 167.
2. Determine which module has the highest overhead. The module naming standard Kxx should indicate the product or component. See Appendix A, "Product codes" on page 241 for internal code names. Figure 5-6 on page 150 shows KDFxxxxx modules that are part of the IBM Tivoli OMEGAMON XE for Storage solution.

```
>                       Display the CPU Usage of a Job

> This screen allows you to see CPU utilization by:  TCB, load module for
> a TCB, CSECT for a load module and offset within a CSECT.
>
> To see CPU utilization by TCB press ENTER.  To see subsequent levels of
> CPU utilization, place the cursor on a specific TCB, load module and
CSECT,
> pressing ENTER after each selection.

 INSP
+    Current sampling options:
+      JOB(CXEGA01)  ASID is 261     Sampling is active
+      SAMPLES(3000)     INTERVAL(10)
+      Samples taken = 1349;  samples used = 11
 icpu  TCB(*)
+<.:  %CPU Utilization (TCB)
|----+---20---+---40---+---60---+---80---+---100
+   1 006FDF30 IEAVAR00    .00 |
+   2 006FD230 IEESB605    .00 |
+   3 006D9E88 KLV         .00 |
...
...
+  45 006A5978 KLV         .00 |
+  46 006A5B28 KLV         .00 |
+  47 006A5CD8 KLV         .00 |
+  48 006A5E88 KLV         .00 |
+  49 006D9188 KLV         .00 |
+  50 006D9338 KLV        5.03 |=>
+  51 00694CB8 KDFSHSMM    .00 |
+  52 006A52D8 KDFSHSML    .00 |
+  53 006D94E8 KLV         .00 |
+  54 006D9698 KLV        5.49 |=>
+  55 006D9848 KLV         .00 |
+  56 006D99F8 KLV         .00 |
+  57 006D9BA8 KLV         .00 |
+  58 006AEE88 KLV       88.92 |=======================================>
+  59 006FD4C0 IEAVTSDT    .00 |
```

Figure 5-6 OMEGAMON II for MVS INSPECT (Classic interface)

3. Disable all the situations associated with the execution of that module.
4. Measure the CPU of the Tivoli Enterprise Monitoring Server without the situations or background data collector started and determine the percentage reduction.

5. Look at the RKLVLOG for KRAIRA000 message for the number of _Z_ situations scheduled for that data collector. Determine whether there is a way to reduce the number of these situations. (See 5.4.5, “Situation synchronization” on page 168).
6. If you still experience problems, it may be a *hidden* situation. Determine which UADVISORs are running.

Isolating using CPU seconds

If you determine that a CPU increase or spike occurs whenever you press Enter, you can estimate the cause by the interval. You are not incurring the overhead from the interval but from user activity at that time.

Use the following procedure if you are unfamiliar with OMEGAMON II for MVS component’s INSPECT facility, the module names cannot be resolved, or it is running on a distributed platform. You must have all of the situations autostarted to maximize situation synchronization efficiency. If you start one situation at a time, none will share data collection with other situations.

The procedure below is to dynamically disable, not enable, situations. Remember to measure CPU at the Tivoli Enterprise Monitoring Agent, remote Tivoli Enterprise Monitoring Server, and hub Tivoli Enterprise Monitoring Server for each of the steps. The following steps provide a high-level overview of the procedure.

> **Note:** If you need to investigate Tivoli Enterprise Monitoring Server address space CPU consumption to this level of detail, it should be done with assistance from an IBM Tivoli IT Specialist, an IBM Software Services for Tivoli (ISST) Consultant, or an IBM Software Support Specialist. Additionally, a Problem Management Record (PMR) should be opened so the issue can be properly tracked and any defects properly documented.

1. With all of the situations running, measure the process / address space CPU for one minute longer than the highest situation interval to ensure that all background collector executions are captured. This is typically six minutes.
2. Disable all situations associated with a given table / probe. Measure the process / address space CPU for one minute longer than the longest remaining situation interval to ensure that all background data collector executions are captured.
3. Calculate the reduction in CPU seconds. Save this number for step 6 on page 152.
4. Repeat steps 2 and 3 until all of the situations are disabled.

5. Measure the process or address space CPU for several minutes. Subtract remaining CPU (not associated with situations) from the CPU number in step 1. This number is total CPU associated with all situations.
6. Convert numbers for each table that were gathered in step 3 on page 151 to a percentage of the situation CPU gathered in step 5.
7. Now that you know which table is the most expensive, verify that all situations written against that table are eligible for situation synchronization.
8. Check the number of synchronized situations that are scheduled for that table. Check that there are no extra situations due to unique sampling intervals. Do this by looking in the agent's RKLVLOG for a KRAIRA000 message for number of _Z_ situations scheduled for that data collector. Try to determine whether there is a way to eliminate any of these, using the Criteria for Combining Situations into One Data Sampling rules.
9. After minimizing the number of unique situations, verify that all are required. Some situations may be overly complex, having more predicates than needed to determine a problem.
10. If you are still experiencing problems, it may be a hidden situation. Find out which UADVISORs are running, as outlined in the next section.

Determine which UADVISORs are running

Browse the RKDSSITF file of the sharing RTE and look in column 119 (field before `*NO, *SYN`) for the UADVISOR situation intervals. The source of these records is stored in the target library in TKANDATV in the member KxxUSITF for same field. Unless you have modified these intervals, those data should still be valid. Figure 5-7 shows the display.

```
 Process   View   Options   Help
--------------------------------------------------------------------------------
 DITTO/ESA for MVS              VB - VSAM Browse

 CATALOG
 RBA 3852332      Key 'Tivoli_NFS_No_Response      '   Col 1        Format CHAR
 VOLSER OMXE3C   Type KSDS   DSNAME OMEGAMON.SHR1.RKDSSITF

 RBA        Len     <===5===10====5===20====5===30=>..5...40....5...50....5...60
 3852332    2092    Tivoli_NFS_No_Response                          No NFS respo
 3858432    2092    Tivoli_Su_Failure                               Switch user
 4325376    3994    Unwanted_inetd_Process         .                Detect unwan
 4329370    1576    UADVISOR_KM5_ASREALSTOR                         Address_Spac
 3866624    3994    UADVISOR_KM5_VCF_CACHE         .                Uadvisor sit
 3870618    3994    UADVISOR_KM5_VCF_CLIENT        .                Uadvisor sit
 3874816    3994    UADVISOR_KM5_VCF_LIST          .                Uadvisor sit
 3883008    3994    UADVISOR_KM5_VCF_PATHS         .                Uadvisor sit
 3891200    3994    UADVISOR_KM5_VCF_STRMVS        .                Uadvisor sit
 3899392    3994    UADVISOR_KM5_VCF_SYSTEM        .                Uadvisor sit
 3907584    3994    UADVISOR_KM5_VDASD             .                Uadvisor sit
 3915776    3994    UADVISOR_KM5_VGLBLENQ          .                Uadvisor sit
 3923968    3994    UADVISOR_KM5_VGRS_TABLE        .                Uadvisor sit
 3932160    3994    UADVISOR_KM5_VWAS              .                Uadvisor sit
 3940352    3994    UADVISOR_KM5_VWASSC            .                Uadvisor sit
 3948544    3994    UADVISOR_KM5_VWPR              .                Uadvisor sit

 Command ===>                                                       Scroll PAGE
 F1=Help  F2=Zoom  F3=Exit  F4=Left  F5=Right  F6=RFind  F7=Bkwd  F8=Fwd
 F10=Actions  F11=CRetrieve  F12=Cancel
```

Figure 5-7   Browsing RKDSSITF for UADVISOR

Going to column 119, you can see the situation similar to Figure 5-8 on page 154.

```
 Process   View   Options   Help
 --------------------------------------------------------------------------------
 DITTO/ESA for MVS              VB - VSAM Browse

 CATALOG
 RBA 3852332      Key 'Tivoli_NFS_No_Response      '   Col 100      Format CHAR
 VOLSER OMXE3C   Type KSDS   DSNAME OMEGAMON.SHR1.RKDSSITF

 RBA        Len   100....5...10....5...20....5...30....5...40....5...50....5....
 3852332    2092               0  001500  *NO
 3858432    2092               0  001500  *NO
 4325376    3994               0  001500  *NO
 4329370    1576               0  006000
 3866624    3994               0  000530  *NO
 3870618    3994               0  000530  *NO
 3874816    3994               0  000530  *NO
 3883008    3994               0  000530  *NO
 3891200    3994               0  000530  *NO
 3899392    3994               0  000530  *NO
 3907584    3994               0  000540  *NO
 3915776    3994               0  000135  *NO
 3923968    3994               0  000135  *NO
 3932160    3994               0  000515  *NO
 3940352    3994               0  000515  *NO
 3948544    3994               0  000515  *NO

 Command ===>                                                     Scroll PAGE
 F1=Help  F2=Zoom  F3=Exit  F4=Left  F5=Right  F6=RFind  F7=Bkwd  F8=Fwd
 F10=Actions  F11=CRetrieve  F12=Cancel
```

Figure 5-8 Getting the UADVISOR's interval

The value 000530 has different meanings for different types of situations.

- A synchronized situation with *SYN means that the situation fires once every five minutes at 30 seconds after the minute. This *after* value is to load balance background data collection.
- For a regular situation with *YES for autostart or *NO for manual start, the value is simply the interval value, and means once every five-and-a-half minutes.

With OMEGAMON DE for z/OS, you can use a custom SQL query against the situation table; it will display the normally hidden UADVISOR situations and their sample interval in REEV_TIME column. Historical UADVISORS all have PDT columns that begin with *IF *HISTRULE nomenclature, so the query in Example 5-2 on page 155 captures the relevant information.

Example 5-2 ODBC query for UADVISOR history situations

```
SELECT SITNAME, REEV_TIME FROM O4SRV.TSITDESC WHERE PDT LIKE '*IF
*HISTRULE*';
```

5.2.4 Disabling products and functions

To isolate the product and functions for finding and correcting performance problem, you can disable background data collectors.

Background data collectors

Some products have background data collectors that are started out of RKANCMD members. Products are often enabled with IRAMAN KxxAgent START commands in KxxSTART or KxxAGST members. Many of these commands can be reversed with a stop command. For example, KXDCMD STOP quiesces the sysplex component of IBM Tivoli OMEGAMON XE for z/OS.

You might be directed by IBM Software Support to issue these commands. BIM Customer Support will provide the command and syntax required. You might be instructed to use this process if you have a storage quiesce or an abend and think it is caused by a particular component. To verify, shut down the specified component. If the problem persists, it is not the terminated component.

Example 5-3 RKANCMD START specification

```
AT ADD ID=KIP DELAY=00:00:15 CMD='IRAMAN KIPAGENT START'
AT ADD ID=KM3 DELAY=00:00:30 CMD='KXDCMD START'  * START EXDC
```

Selective load

You can effectively disable a product or component by selectively *not* loading the code that provides that product or component.

> **Note:** Modify the load lists for any IBM Tivoli OMEGAMON XE solution *only* under the direction of IBM Software Support. Due to the interactions of various products and various components, there can be unintended results.

Each z/OS started task, either Tivoli Enterprise Monitoring Server or Tivoli Enterprise Monitoring Agent, has a KxxSYSIN member in RKANPAR that lists all products that are to be loaded. To disable an entire product, comment out its LOADLIST statement. For example, the Tivoli Enterprise Monitoring Server has a member in RKANPAR called KDSSYSIN, which has entries like LOADLIST(KOSLLIST). This particular entry loads the IBM Tivoli OMEGAMON XE for z/OS sysplex component modules.

- To disable the entire product:

  Edit RKANPAR(KxxSYSIN), where xx is your product or address space code, and comment out the load of product code *LOADLIST(KxxLLIST) for the entire product.

- To disable a product component:

  Edit RKANPAR(KxxLLIST) and comment out the modules in question. Be aware that modules may be in more than one LOADLIST. For example, IBM Tivoli OMEGAMON XE for z/OS and IBM Tivoli OMEGAMON XE for Storage both use the shared DASD data collector and therefore will have LOADLIST statements for this component.

IBM Tivoli OMEGAMON XE for IMS on z/OS have commands for disabling and starting collectors. To stop collection permanently for an entire IMS, DB2, or IRLM subsystem, remove the command from the KxxSTART member in RKANCMD. Collection can be disabled temporarily without recycling the Tivoli Enterprise Monitoring Server to aid in problem or overhead determination. The modify command in Example 5-4 will stop all IMS collectors.

Example 5-4 Disable IMS collectors

```
F stcname,IPDC STOP IRLM(irlmstcbrlm)
F stcname,IPDC STOP IMS(imsssidi51b)
```

It is possible to alter or disable selective collectors while the Tivoli Enterprise Monitoring Server is running. Change the sample interval to zero for the component in RKANPAR. Then issue the previously noted stop commands followed by the start commands in Example 5-5.

Example 5-5 Start IMS collectors

```
F stcname,IPDC START IRLM(irlmstcbrlm)
F stcname,IPDC START IMS(imsssidi51b)
```

IBM Tivoli OMEGAMON XE for DB2 on z/OS has these commands for disabling and starting collectors. To stop collection permanently for an entire DB2 or IRLM subsystem, remove the command from the KxxSTART member in RKANCMD. Collection can be disabled temporarily without recycling the Tivoli Enterprise Monitoring Server to aid in problem or overhead determination. The MODIFY commands in Example 5-6 stop all DB2 collectors.

Example 5-6 Disable DB2 collectors

```
F stcname, DPDC STOP IRLM(irlmstcbrlm)
F stcname, DPDC STOP DB2(db2ssidd411)
```

It is possible to alter or disable selective collectors while the Tivoli Enterprise Monitoring Server is running. Change the sample interval to zero for the component in RKANPAR. Then issue the stop commands in Example 5-6 on page 156, followed by the start commands in Example 5-7.

Example 5-7 Start DB2 collectors

```
F stcname,DPDC START IRLM(irlmstcbrlm)
F stcname,DPDC START DB2(db2ssidd411)
```

Disabling UADVISORs

Normally a product-provided or user-defined situation has to be active for data collection to occur. However, invisible situations known as UADVISORs can drive data collection.

> **Note:** UADVISORs should be disabled only under the direction of IBM Software Support. Due to the interactions of various products and various components, there can be unintended results. Remember that maintenance may cause re-seeding and re-activation of the UADVISOR situations.

To disable UADVISOR situations:

- Do not seed the UADVISOR situation initially. Change TKANDATV(KxxSITx) before seed.
- Delete UADVISOR from the KDSSITF file.
- Execute SQL to change the *SYN (the autostart for the UADVISORS) to *NO. You can change back again when you want to re-enable them.

When you disable UADVISOR situations, do so on the hub Tivoli Enterprise Monitoring Server and all z/OS-based remote Tivoli Enterprise Monitoring Servers. The Tivoli Enterprise Monitoring Servers must be recycled for the change to take effect. The SQL code in Example 5-8 changes the autostart flag and must be issued before the recycle. The next restart will honor the changed setting.

Example 5-8 SQL to update AUTOSTART field

```
SELECT SITNAME, AUTOSTART
FROM O4SRV.TSITDESC WHERE SITNAME LIKE 'UADVISOR_OMSYS_VDASD';

UPDATE O4SRV.TSITDESC SET AUTOSTART = '*NO' /* Change to *SYN TO
Re-enable*/
WHERE SITNAME LIKE 'UADVISOR_OMSYS_VDASD';
```

```
SELECT SITNAME, AUTOSTART
FROM O4SRV.TSITDESC WHERE SITNAME LIKE 'UADVISOR_OMSYS_VDASD';
```

To run the SQL in Example 5-8 on page 157, create a new member in RKANSQL that will be invoked using the SPUFI modify command as shown in Example 5-9 to the Tivoli Enterprise Monitoring Server started task. Verify that the modify produces a NO on the second display and that the situation does not start on the recycle.

Example 5-9 Modify command to drive SPUFI updates

```
F stcname,CTDS START SPUFIL,mbrname
```

Create separate members to control different situations. The Tivoli Enterprise Monitoring Server started task RKLVLOG will have the results of the command.

5.3 Query optimization and tuning

This section discusses = tuning the queries that are processed to display metrics—be they tabular, charts, or graphs—within Tivoli Enterprise Portal workspace views.

5.3.1 Query processing

The query assigned to a chart or table view requests data from a particular attribute group. It executes when you open or refresh the workspace. Queries make up the processing load for on-demand data collection. You can reduce the frequency and amount of data sampling by:

- Customizing the query to filter out unwanted data. This reduces the number of selection criteria (rows) and attributes (columns) collected.
- Applying the same query to other views in the workspace. This reduces the number of data samples required: one query uses a single sample for multiple views.
- Adjusting automatic refresh to longer intervals. This causes Tivoli Enterprise Monitoring Agent data to be collected less frequently. A graphic view workspace uses a significantly less data compared to a table view.

> **Note:** Do not confuse custom queries with view filters from the Filters tab of the query Properties editor. View filters fine-tune the data after it has been retrieved by the query and do not reduce network traffic, data collection processing, or memory demands.

The following general recommendations and observations should be considered as well:

- Some attributes are more expensive to retrieve than others. One *expensive* column in a table will make any workspace view or situation that references that table more expensive. An example of an expensive attribute is one that must run long storage chains to determine its value, such as using a process table to look for looping tasks.
- Column function (such as MIN, MAX, AVG, and so on) requires post processing of the query answer set after data is returned to Tivoli Enterprise Monitoring Server.
- Use more efficient data manipulating functions, such as substring instead of string scan. If you know the position of the string to search, do not scan the whole string to check for the value.

5.3.2 Defining custom queries

Custom queries reduce network traffic, processing at the agent and Tivoli Enterprise Monitoring Server, and memory usage at the Tivoli Enterprise Portal Server and Tivoli Enterprise Portal client. Custom queries accomplish this by limiting the number of rows and columns passed from the Tivoli Enterprise Monitoring Agent to the Tivoli Enterprise Monitoring Server.

Most of the predefined, product-provided queries request all columns and all rows, of which only a few may be of interest to you. Removing the unwanted columns (or attributes) may reduce another 25% of the characters. By eliminating the characters you eliminate the need to translate them, to store them in memory, and to transmit them from Tivoli Enterprise Monitoring Agent to Tivoli Enterprise Monitoring Server to Tivoli Enterprise Portal Server.

It is recommended to tune (the queries servicing) workspaces that are frequently executed, return large quantities of metrics, or both, because auto-refreshing always requires resources, and intermittent extremely large reports cause a spike in memory requirements.

- Restricting the number of rows

  Most predefined queries return all rows and columns. You can create a custom query to filter out the irrelevant or uninteresting metrics. Not only does this make it easier to read the report, but it saves Tivoli Enterprise Portal Server memory, client memory, and CPU because there are fewer rows to translate, sort, and transmit.

  You may be using view filters inappropriately. View filters work only on the current page returned from the query. For example, if page size is 100 lines and the filter reduces it to five lines on page 1 and similarly on subsequent pages, the row set cannot be seen on one page. Do not increase the workspace page size to see everything on one page. Increased page size actually increases Tivoli Enterprise Portal client memory requirements. Instead, avoid this condition by creating a custom query that filters the data at query execution time.

- Restricting the number of columns

  Most predefined queries return all columns (or attributes). A particular attribute group may contain 50 columns, yet all you need is five. Creating a custom query to retrieve just the desired five attributes will reduce Tivoli Enterprise Portal Server and client CPU and memory.

  The product-provided DASD_MVS_Devices query returns 32 columns for approximately 5,100 rows of data in the ITSO environment. Many of the default columns and many of the rows of data may not be of interest. The modified version of the query as shown in Figure 5-9 on page 161 removes unwanted columns and selects only devices with an I/O rate greater than zero and contains only 17 columns and 20 rows of data—a drastically smaller amount of data.

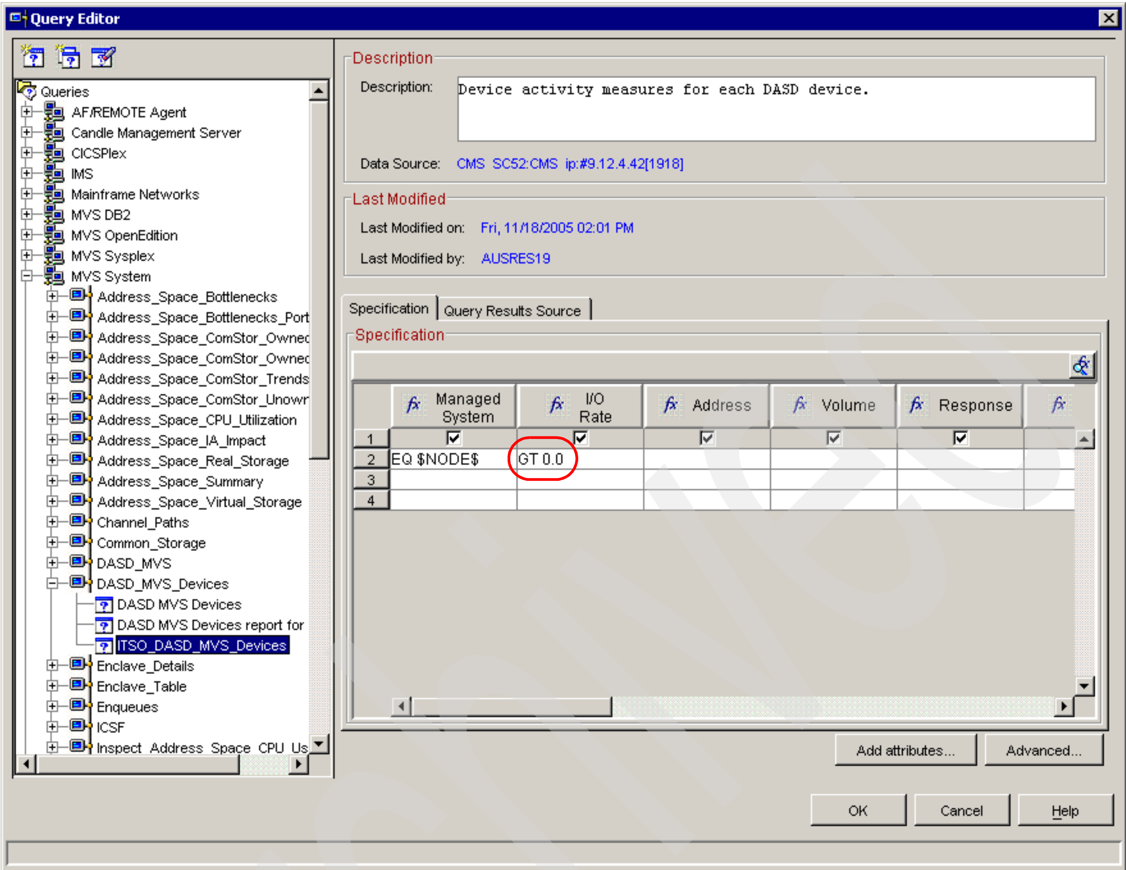


Figure 5-9 ITSO DASD MVS Devices query

- Use the same query in a workspace

  If you have multiple views in a workspace with different data from different attribute groups, you will need a different query for each group. But if the views have data from the same attribute group, use one query to accommodate both. Two unique queries will drive data collection at the agent and increase data collection overhead. For each workspace, have one query that can be shared by all views using that attribute group. Remember that the entire results set for each query is stored on the Tivoli Enterprise Portal Server, so avoid duplicate result sets.

- Collect agent data less frequently

  A good practice is to not use auto refreshing. The Navigator view and event views (message log, event console, and graphic console) will refresh automatically. This provides you with instantaneous alerts, and you can navigate to their event workspaces with actual data. The graphic view, the graphical equivalent of the Navigator, which shows alerts but no data, affects client memory but not the Tivoli Enterprise Portal Server.

5.4 Tuning situations

Situation is the mechanism in IBM Tivoli OMEGAMON XE that generates an alert when an out-of-norm condition occurs. The situation evaluates system attributes and decides whether a potential problem condition is occurring. The situations are evaluated on a regular interval. This section covers analyzing situation processing and reducing its processing requirements.

5.4.1 Situation processing

Situations are a mechanism in IBM Tivoli OMEGAMON XE that automatically performs monitoring for you. Situations collect performance indicators and subject them to conditional checking, which is associated with a threshold. When the condition is true, the situation is raised or fires and an event is generated. These state changes are passed to the Tivoli Enterprise Monitoring Server.

The frequency of state changes affects hub and remote Tivoli Enterprise Monitoring Server capacity. It also affects storage requirements because these events are saved in memory and written to disk. The sampling intervals themselves affect Tivoli Enterprise Monitoring Server capacity.

Several important considerations for situation processing that can greatly affect the performance of the IBM Tivoli OMEGAMON XE system are discussed here:

- Situation distribution
- Situation predicate processing
- Situation threshold and interval
- Situation synchronization
- Situation and policies
- Embedded situations

5.4.2 Situation distribution

All situations are in RKDSSITF, a VSAM file in z/OS-based Tivoli Enterprise Monitoring Server, or the QA1CSITF file on a distributed-based Tivoli Enterprise Monitoring Server. This file shows every user-defined situation, including those that are not distributed or are distributed but not started.

Predefined situations are in the TKANDATV(KxxSITx) file. It shows where the situations originated when you registered or seeded the Tivoli Enterprise Monitoring Server.

Situations can be autostarted or manually started. Situations assigned to managed systems, autostarted or not, are distributed to their Tivoli Enterprise Monitoring Agents at Tivoli Enterprise Monitoring Server startup time. This requires processing and affects Tivoli Enterprise Monitoring Server startup time.

If a Managed System List (MSL) is updated, then all the situations that use the MSL are redistributed to all managed systems in the list that require processing at the Tivoli Enterprise Monitoring Server and the affected Tivoli Enterprise Monitoring Agents. For example, when a new CICS region comes online for the first time, the *CICS Managed System List is updated. If the default distribution assignments have not been modified, this will cause all 272 product-provided CICS situations to be redistributed to all Tivoli Enterprise Monitoring Agents that are monitoring CICS regions. Therefore, you should regularly remove the distribution assignment from any situation that is not autostarted or manually started.

When you copy predefined situations and give them site-specific names to protect them from changes that IBM Tivoli might make in future releases, make sure you disable the original situation by turning off autostart and removing its distribution list. This is critical for many distributed products that distribute all of their predefined situations.

At Tivoli Enterprise Monitoring Server startup, autostarted situations must be distributed to all of their agents. Note that subnodes are not unique distributions. Because situations also use CPU at the agents, Tivoli Enterprise Monitoring Servers that have agents installed on them are also affected.

To stop a situation from autostarting, open the situation editor and select the **Condition** tab, which is the default. If it is checked, remove the check mark from the **Run as startup** specification as shown in Figure 5-10 on page 164. This does not stop a situation that is currently started as it will not take effect until the next startup of the affected Tivoli Enterprise Monitoring Agent. If required, manually stop the situations.

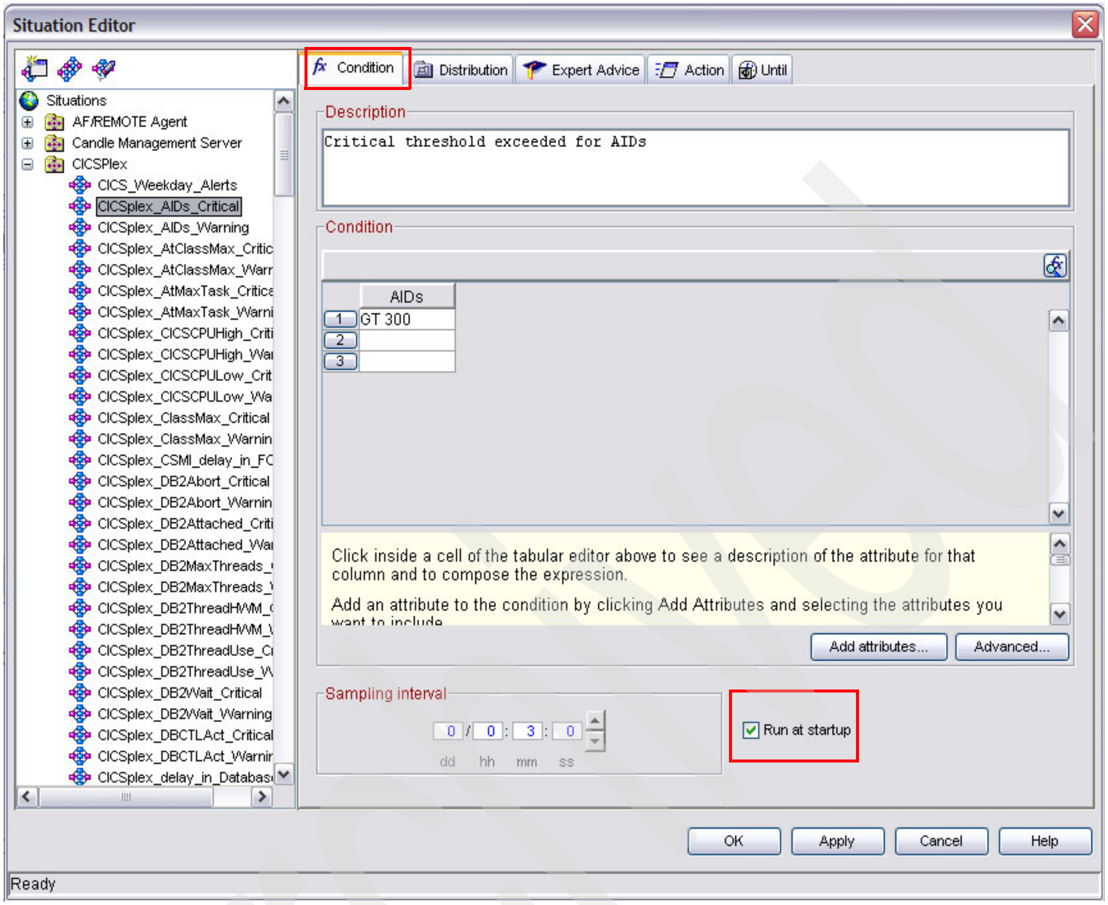


Figure 5-10 Modify situation's Run at startup specification

To modify a situation's distribution, open the situation editor and select the **Distribution** tab. If it is checked, remove the check mark from the **Run as startup** specification. Select the Managed Systems, Managed System Lists, or both in the Assigned window and then use the Add and Remove arrows to remove the assigned managed systems. At the next Tivoli Enterprise Monitoring Agent startup, this situation will not be distributed.

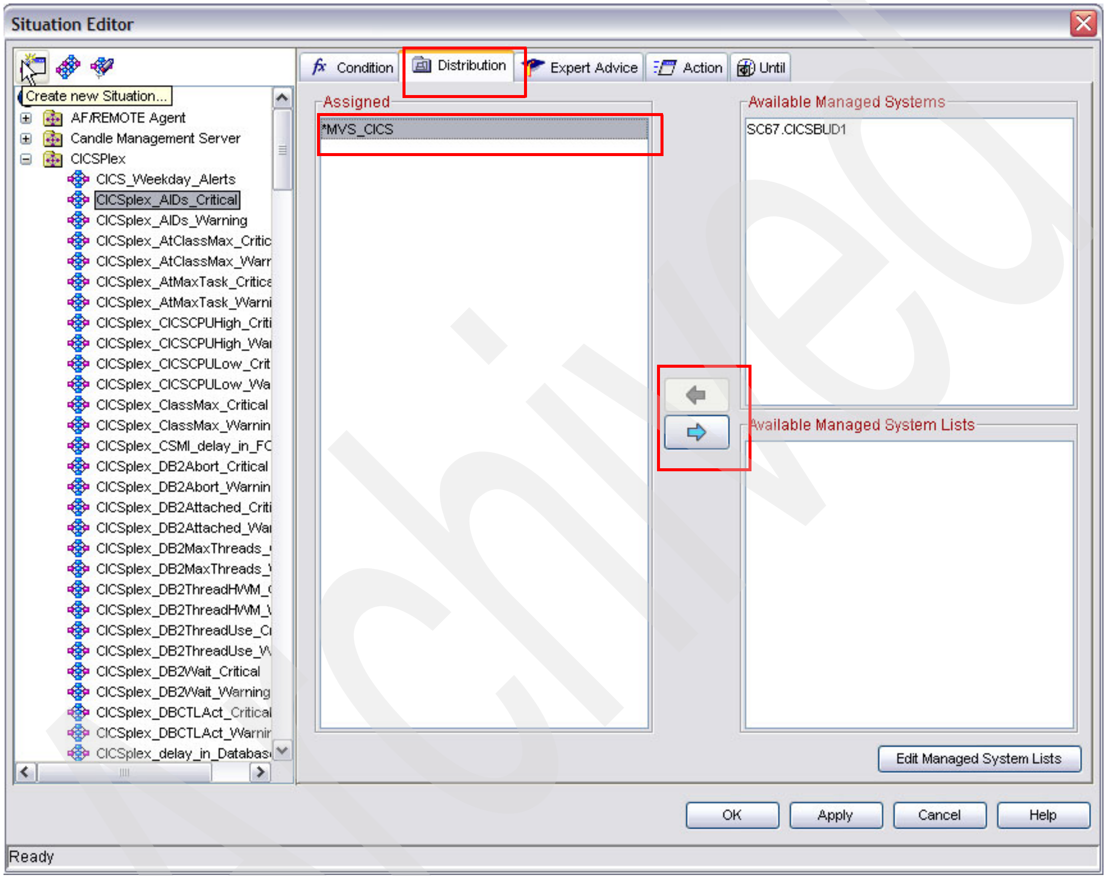


Figure 5-11 Modify situation distribution

Subsystem monitoring

IBM Tivoli OMEGAMON XE will automatically self-discover your CICS regions and DB2 subsystems. It also automatically creates Managed System Lists (MSLs) for situations to monitor these systems - *CICS and *DB2, respectively.

If you previously were not proactively monitoring exceptions using OMEGAVIEW or with logged-on users in auto-update mode, you now have new data collection and processing. OMEGAMON for CICS on z/OS and OMEGAMON for DB2 PM / PE on z/OS can be configured to selectively discover subsystems, which would eliminate this issue. But that would also eliminate the ability to reactively view data from unmonitored test systems. This may be desirable as it also eliminates these subsystems and their associated navigator views, which may clutter your view of your enterprise.

Both of these products can be configured to discover subsystems selectively, which will reduce the processing requirements. This may be desirable because it also keeps nodes off the Navigator Physical view. However, it would also eliminate the ability to view data reactively from unmonitored test systems.

Many customers configure production regions and subsystems to a production hub Tivoli Enterprise Monitoring Server and Tivoli Enterprise Portal Server, and test regions or subsystems to a separate set. If you have systems for which you do not want to perform exception analysis or collect historical data, you should create user-defined distribution lists such as CICS_Production, which will have the list of regions you want monitored.

5.4.3 Situation predicate processing

The situation contains predicates, in which the monitoring attributes to be evaluated are specified. Different predicates generate different processing requirements. It is important to understand how the predicates are processed to analyze these predicates.

Predicates are processed sequentially from first to the last. If you have multiple predicates in a situation, order them to eliminate as many rows as possible as early as possible with the least expensive or most restrictive filter as the first predicate.

Here are some predicate processing tips:

- Numeric attributes are processed more quickly than text attributes.
- String checking with substring (STR) is more efficient than the string scan (SCAN), especially for long strings. If you know the exact location of the text or characters to be evaluated, use a substring.
- A column function is evaluated at the Tivoli Enterprise Monitoring Server, so all available rows will be sent back to Tivoli Enterprise Monitoring Server from the agent, with high network usage, memory, and processing requirements as a result. Try to avoid the predicates such as MAX, MIN, AVG on a column.

5.4.4 Situation threshold and interval

IBM Tivoli supplies a set of default thresholds with each IBM Tivoli OMEGAMON XE solutions. However, the defaults may not be appropriate for your installation. Adjust the thresholds for each exception to fit the performance standards at your installation. For example, situations that are part of a *health check* process can use a longer interval such as 30 minutes, whereas situations that detect a *resource shortage* condition probably need a more frequent interval.

Some IBM Tivoli OMEGAMON XE products have specific critical threshold situations for stall and crash prevention that automatically start by default, but this may not be true for their warning situations. At many sites the warning yellow lights are ignored, so it would be a waste of processing to distribute and start these situations. Other sites pay close attention to warning alerts, in which case the situations should be autostarted. With IBM Tivoli OMEGAMON XE you can set different intervals for every threshold. Be aware that picking different intervals will cause higher overhead. Some products, such as IBM Tivoli OMEGAMON XE for CICS on z/OS, have preselected critical threshold situations for stall and crash prevention automatically started as a default. However, they do not autostart any warning situations.

The predefined situations may have more frequent sampling than required by your environment. Recognize that not all data is equally important, nor is it equally expensive to collect. Some higher processing exceptions may not be crucial to availability in your environment. Ideally, you would like to realize huge benefits at low cost, but this is not always possible. Avoid high processing requirements and low benefits. The answers may be different for various applications running under the same subsystem.

IBM Tivoli OMEGAMON XE attempts to have low overhead defaults. However, no vendor can know your environment or what is relevant to your business applications. IBM Tivoli OMEGAMON XE was designed to provide in-depth analysis, as you will need these capabilities to deal with problems when they have been identified. However, awareness of situations is not necessarily minimized by selectively disabling data collection. It is best to disable the items that you can categorize as low benefit with high overhead.

For migration purposes, IBM Tivoli OMEGAMON XE for *x* on z/OS uses the same default thresholds as the associated OMEGAMON II component. However, you can take advantage of new capabilities to selectively disable Warning thresholds. Follow the instructions in the configuration and customization guide for your IBM Tivoli OMEGAMON XE product to run KXEMIGR, a utility that migrates OMEGAMON II critical and, optionally, warning thresholds to be used as predefined situations in IBM Tivoli OMEGAMON XE.

In some cases, disabling a situation is not an option if it provides high benefit even at a higher processing level. Such situations should be tuned or substituted. Some products or subsystems may have more than one way to detect the same problem and not all situations have the same processing requirement. IBM Tivoli OMEGAMON XE provides Boolean logic and multiple ways to detect complex situations. For example, a CICS transaction rate of zero with high CPU indicates that a single transaction is consuming cycles and not allowing the dispatcher to schedule other transactions. This particular CICS-related situation can be replaced by evaluating the CICS time-of-day clock (TOD) not being updated. Numerous data collection parameters can be adjusted in scope and frequency to reduce overhead.

5.4.5 Situation synchronization

One of the biggest savings in situation processing is to perform *situation synchronization*, or sometimes also referenced as *duperization*. Situation synchronization enables multiple situations to collectively perform a single data collection or take a sample, and the result set is evaluated from a single data sample.

Situation synchronization enables more situations to be evaluated with a constant processing requirement. To be eligible for situation synchronization, the following conditions must be fulfilled. Table 5-1 lists the situation requirements.

Table 5-1 Synchronization eligibility requirements

Situation definition	Needed
Same attribute group required	Yes
Same time interval required	Yes
Can be restarted	Yes
Must be autostarted (except UADVISOR cannot have autostart set to SYN)	Yes
New / update situation supported (requires Tivoli Enterprise Monitoring Server to be recycled)	Yes
Can combine situations with different distribution lists	Yes
Maximum of 10 expressions per situation	Yes
Display Item (atomize) supported	No
Take action allowed	No
Until clause permitted	No

Situation definition	Needed
Can use *Check for Missing Items* function	No
Can be used in a policy	CT/350 - Yes CT/360 - Yes ITM 6.1 - No
Can use STR (return a subset of string) or SCAN (scan for string within string) function	No
Can use group functions: AVG (average), COUNT (count), MAX (maximum), MIN (minimum), SUM (sum)	No
JOIN allowed between two single-row attribute groups	No
Event persistent situations permitted	No

Based on the requirements in the table, the following common principles apply:

- It is not advisable to arbitrarily change the evaluation interval of situations as this may hinder the situation synchronization processing. Setting a longer interval does not necessarily mean that processing requirements are reduced. When you change the interval of one situation, you schedule a new sampling and double the processing. If you take a third situation and set it to yet another interval, you could triple the overhead.
- Some complex situations cannot be synchronized because the predicate length exceeds the record length in RKDSRULD. The RKDSRULD file contains the situation synchronization rules. The actual limitation is the size of the rule. Typically, eight to 10 conditions can be built into a single scheduled sampling, and anything that was previously combined is uncombined when you changed it. Note that some Tivoli Enterprise Monitoring Agents are intelligent enough to automatically collect information once for multiple autostarted situations with the same interval.

To determine whether a situation has been included in synchronization, you can browse the RKDSRULD dataset on the Tivoli Enterprise Monitoring Server that the Tivoli Enterprise Monitoring Agent is connected to. Figure 5-12 on page 170 shows the RKDSRULD file for z/OS.

```
DITTO/ESA for MVS              VB - VSAM Browse

CATALOG
RBA 262144       Key _Z_CICSROV1___________________   Col 1        Format CHAR
VOLSER OMXE3C   Type KSDS   DSNAME OMEGAMON.SHR3.RKDSRULD

RBA        Len     <===5===10====5===20====5===30=>..5...40....5...50....5...60
262144     3032    _Z_CICSROV1_____________________..SELECT CICSROV.AIDS, CICSR
265176     3032    _Z_CICSROV3_____________________..SELECT CICSROV.AIDS, CICSR
268208     3032    _Z_CICSSTOR0____________________.aSELECT CICSSTOR.AREA, CICS
271240     3032    _Z_ICSF2________________________..SELECT ICSF.AVGWAIT, ICSF.
274272     3032    _Z_ICSF4________________________..SELECT ICSF.AVGWAIT, ICSF.
277304     3032    Crypto_CKDS_Access_Disabled_____.nSELECT ICSF.AVGWAIT, ICSF.
0          3032    Crypto_CKDS_80PCT_Full__________.~SELECT ICSF.AVGWAIT, ICSF.
3032       3032    Crypto_Internal_Error___________..SELECT ICSF.AVGWAIT, ICSF.
6064       3032    Crypto_Invalid_Master_Key_______.~SELECT ICSF.AVGWAIT, ICSF.
9096       3032    Crypto_Invalid_PKA_Master_Keys__.1SELECT ICSF.AVGWAIT, ICSF.
12128      3032    Crypto_No_Coprocessors__________..SELECT ICSF.AVGWAIT, ICSF.
```

Figure 5-12 RKDSRULD dataset contents

Figure 5-12 contains the definition that is being sent to the Tivoli Enterprise Monitoring Agents, including predefined and custom situations. Any duperized situations appear at the beginning of the file, with names start with _Z_.

Looking at the list of synchronized situations, you may see several _Z_ entries with different suffixes. These are the different synchronized collections that are performed for that data source. The example in Figure 5-12 shows two synchronized collections of CICSROV the table. This can be the result of too many predicates for one situation or of different collection intervals.

You can also check the messages from the RKLVLOG message, KRAIRA000, that indicate which synchronized (duperized) situations are being started. Example 5-10 shows a sample message.

Example 5-10 RKLVLOG message for duper situation

```
KRAIRA000, Starting _Z_ICSF4 <3145945,1048781> for KM5.ICSF.,
Producer(IRA Constructor)
```

5.4.6 Situation and policies

Policies can be a significant source of processing in situation overhead. Policies can more than double the processing requirements of collecting data for situations. This is because they duplicate the collection for the situation to drive alerts to the Tivoli Enterprise Portal Server and are not eligible for situation synchronization.

If you have three situations that are eligible for synchronization in one synchronized data collection, and you add them to policies, the result will be additional schedulings of the collector. Depending on the level of the code, the behaviors are:

- Level 350 and level 360: Policies have no impact on synchronization eligibility. A policy-based situation is always a separate schedule and the second schedule is never eligible for synchronization. The original situation is eligible for synchronization. Level 350 is shipped with the IBM Tivoli OMEGAMON XE on z/OS V100 or V220. Level 360 is shipped with the IBM Tivoli OMEGAMON XE on z/OS V3.1.0.
- IBM Tivoli Monitoring V6.1: Makes the original situation not eligible for synchronization but shares the collected data with the policy. This offers some processing reduction compared to previous levels. ITMS:Engine V400 is shipped with the IBM Tivoli Monitoring V6.1 solutions and can be enabled to support IBM Tivoli OMEGAMON XE on z/OS V3.1.0.

In a policy, the `Evaluate a Situation Now` activity does not actually start the situation, it just performs one-time sampling of data; however, `Wait Until A Situation is True` activity does start the situation.

5.4.7 Embedded situations

Embedded situations are inserted into the formula of another situation, called the *embedded* or *parent* situation. The properties of the parent situation override those of the embedded situation. Thus, the parent situation uses its interval to drive a take sample for the data required from the embedded situation. You might use embedded situations to look for time-dependent workload issues.

The embedded situation runs independent from the original (non-embedded) form. When the same situation is embedded in several different other situations, they are triggered independent of each other, which can cause excessive situation evaluation and unnecessary processing. However, when embedding several product-specific situations into a single parent situation, each situation runs only once (as long as none of the embedded situations are autostarted).

An example of embedded situation usage is to provide a time-sensitive situation. Instead of changing all situations to include time checking, you can use embedded situations. There are two approaches to this. The following example illustrates the differentiation of weekday and weekend situations:

- Create a common Weekday situation that detects the day of the week to be greater than or equal to 2 and less than 6 (Monday to Friday). Now embed all weekday-sensitive situations into the Weekday situation. This means that if you have 20 situations embedded in the Weekday situation, then the Weekday situation will be executed 21 times if Weekday is autostarted. This can mean unnecessary processing.

- Another alternative is to have the Weekday situation embed all situations that must be evaluated on a weekday. This removes the overhead of the additional 20 Weekday situations evaluation. However, the generated event will be a generic one, called Weekday. (You could use a more meaningful name for Weekday, such as CICS_Weekday_Alerts, as shown in Figure 5-13). This situation checks to see whether the day of the week is Monday through Friday, and if true, it evaluates the situation CICSplex_AtMaxTask_Critical.

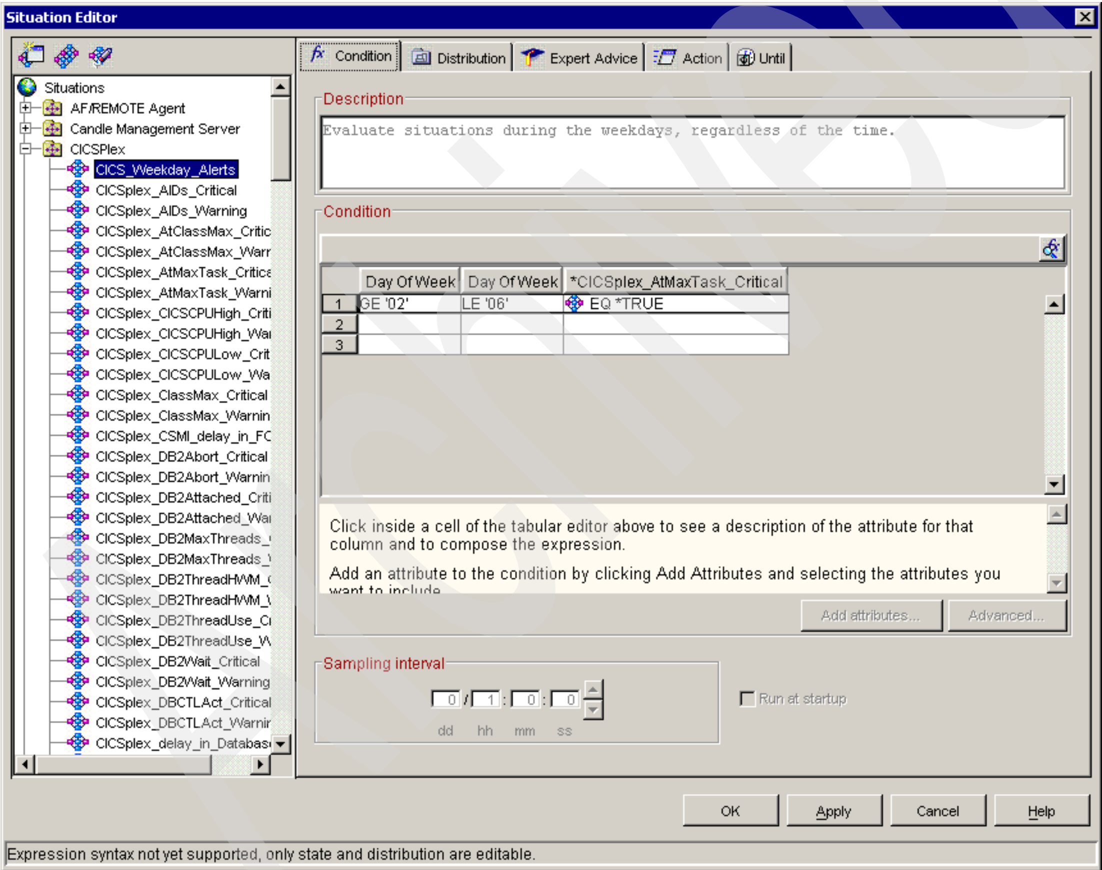


Figure 5-13 Embedded situation - CICS_Weedkday_Alerts

Any time you use embedded situations, neither the parent nor child are eligible for situation synchronization. This may cause the product-specific situation to run more often and increase the processing requirements for the additional data collections that are being scheduled.

An alternative for reducing the required processing for time-sensitive alerts is to create a policy to control false alerts based on time. This has its benefits in that all situations will be eligible for situation synchronization and the Tivoli Enterprise Monitoring Server situation monitor will not have to process (filter out) alerts during the time frame when they are not applicable. Having all situations synchronized will reduce the total number of situations, which will also reduce Tivoli Enterprise Monitoring Server and agent overhead.

These steps outline the procedure for coding time policies to reduce the agent and Tivoli Enterprise Monitoring Server processing requirements:

1. Create two situations that check the day of the week.
   - Weekday situation: check for weekday DAYOFWEEK>=2 and DAYOFWEEK<=6
   - Weekend situation: checking for weekend DAYOFWEEK=1 or DAYOFWEEK=7
2. Create two policies:
   - Weekday_Policy that, based on the active date situation (Weekday), will restart all the situations that should be active.
   - Weekend_Policy policy that, based on the inactive period, will stop all the situations you started in the first policy.

   By default, policies are set to autostart. Be sure to maintain that setting for both of these policies. Do not rely on the policy to start the situations the first time, as they will not be eligible for situation synchronization processing unless they are autostarted. See Figure 5-14 on page 174 for an example of the Policy definition.

   In this figure, the policy executes when situation Weekday_Starter_Situation is true. At that time, the policy will start four situations.

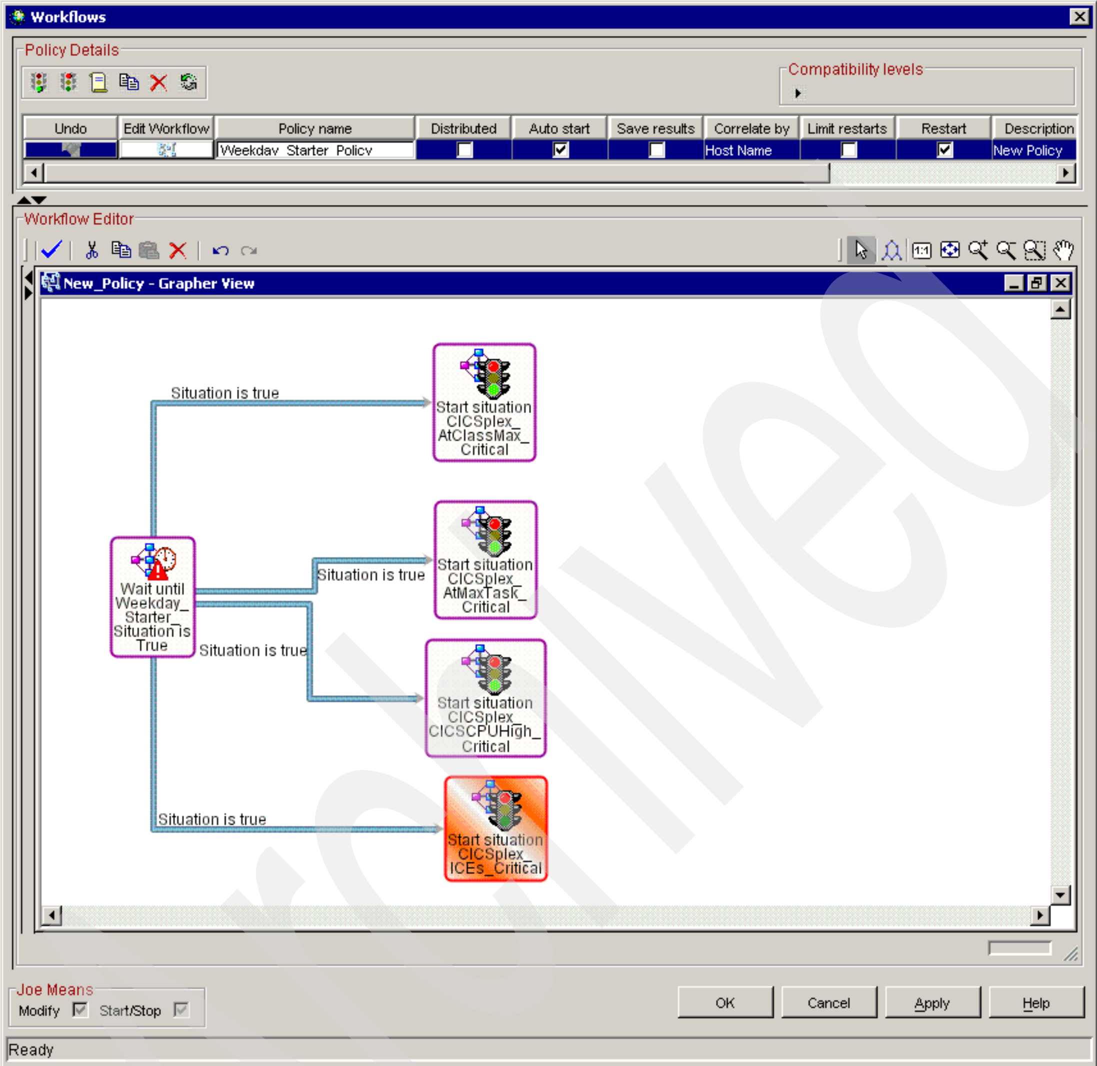


Figure 5-14 Policy to start situations based on day-of-week

3. Make sure that all product situations are autostarted so they will become synchronized. You do not need to start the time/date dependent situations, as that will be done automatically by policies. This scenario does not have embedded situations that are automatically started by the parent situation.

This same concept can be useful for eliminating false alerts for scheduled maintenance windows. For example, if all CICS regions are brought down every night from 2:00 a.m. until 4:00 a.m., you want to avoid the alerts associated with this planned event. You achieve this by creating a Begin_Maintenance_Window policy with processes to stop all situations that generate false alerts. You would also have a corresponding End_Maintenance_Window policy. These policies could be manually started and stopped by the operator instead of acknowledging all of the alerts that are normally generated.

5.5 Product-specific tuning

This section provides some product-specific tuning information for IBM Tivoli OMEGAMON XE products:

- 5.5.1, "IBM Tivoli OMEGAMON XE for CICS on z/OS" on page 175
- 5.5.2, "IBM Tivoli OMEGAMON XE for z/OS" on page 180
- 5.5.3, "IBM Tivoli OMEGAMON DE on z/OS" on page 186
- 5.5.4, "IBM Tivoli OMEGAMON XE for IMS on z/OS" on page 188
- 5.5.5, "IBM Tivoli OMEGAMON XE for DB2 on z/OS" on page 190

5.5.1 IBM Tivoli OMEGAMON XE for CICS on z/OS

Many sites have significantly more CICS regions to monitor than z/OS LPARs, DB2 subsystems, or IMS control regions. For this reason, the CICS environments deserve attention when tuning. By default, all your CICS regions will be auto-discovered if IBM Tivoli OMEGAMON XE for CICS on z/OS is installed on an z/OS partition.

We discuss the following issues with IBM Tivoli OMEGAMON XE for CICS on z/OS:

- Situation synchronization
- Excessive number of situations to create exceptions
- Auto-discovery of CICS regions
- Situation distribution
- Historical data collection

Situation synchronization

To reduce IBM Tivoli OMEGAMON XE for CICS on z/OS processing requirements and make it comparable to the OMEGAMON II for CICS component, the default situation intervals match the OMEGAMON II alert frequencies. However, in the case of the CICS Region Overview table, there are five unique frequencies that drive five executions of the Region Overview data collector. By changing the situation intervals to a common value, we can reduce the frequency of data collection by making the situations eligible for situation synchronization. Table 5-2 lists situations whose intervals should be reduced to three minutes, which would match another set of situations that are being collected from the same table.

Table 5-2 CICS Region Overview situations

Situation name	Interval
CICSplex_CICSCPUHigh_Critical	0:00:04:00
CICSplex_CICSCPUHigh_Warning	0:00:04:00
CICSplex_CICSCPULow_Critical	0:00:04:00
CICSplex_CICSCPULow_Warning	0:00:04:00
CICSplex_IORateHigh_Critical	0:00:04:00
CICSplex_IORateHigh_Warning	0:00:04:00
CICSplex_PageRate_Critical	0:00:05:00
CICSplex_PageRate_Warning	0:00:05:00
CICSplex_TranRateHigh_Critical	0:00:10:00
CICSplex_TranRateHigh_Warning	0:00:10:00
CICSplex_TranRateLow_Critical	0:00:10:00
CICSplex_TranRateLow_Warning	0:00:10:00

Excessive number of situations to create exceptions

IBM Tivoli OMEGAMON XE for CICS on z/OS distributes situations that enable you to duplicate any of the exceptions that were being monitored by the OMEGAMON II for CICS CUA interface. Most of these situations are not autostarted. Use caution when enabling these situations to avoid performing duplicate or redundant monitoring. For example, many conditions are either true or false in the OMEGAMON II for CICS CUA but you were asked to assign Warning or Critical status for these conditions. In IBM Tivoli OMEGAMON XE for CICS on z/OS, there is no mechanism that prevents you from enabling both the Warning and Critical situations for these and both would fire simultaneously, which would cause additional processing for evaluation and when the situations evaluated true.

Distributing and evaluating situations requires processing. For example, you may have the same thresholds for all of your LSR pools. If you use only three pools but distribute and start the situations for all eight pools, the result will be additional processing. You could create new situations that do not reference the pool name, thereby applying the same thresholds to all pools and reducing the number of situations. You would then want to ensure that the original 48 situations (three critical and three warning situations per each of eight LSR pools for a total of 48 situations) were not started or distributed to CICS regions.

Auto-discovery of CICS regions

IBM Tivoli OMEGAMON XE for CICS on z/OS automatically discovers your CICS regions to build the Navigator Tree Physical view. It also automatically assigns the *CICS distribution list to situations to monitor these systems. This could introduce some unnecessary processing, as you may not want to monitor all the regions. In addition, the more members in the distribution list, the longer it can take to initialize the Tivoli Enterprise Monitoring Server for situation processing.

IBM Tivoli OMEGAMON XE for CICS on z/OS can be configured to discover subsystems selectively, which eliminates this issue. This can also eliminate the ability to view data from these unmonitored systems at Tivoli Enterprise Portal. This might be desirable because it also keeps them from cluttering the Physical Navigator view.

You may configure production regions to a production hub Tivoli Enterprise Monitoring Server and Tivoli Enterprise Portal Server, and test regions to a separate set. To avoid performing exception analysis on all regions, or collecting historical data for them, do not use the *CICS distribution list. Instead, create a new managed system list for the regions you want to monitor, such as CICS_Production.

The facility to edit Managed System Lists is available from multiple start points, such as the Edit option on the Action Bar (Figure 5-15). From the Edit Manage System List pop-up you can create a new MSL or create a copy of an existing MSL.

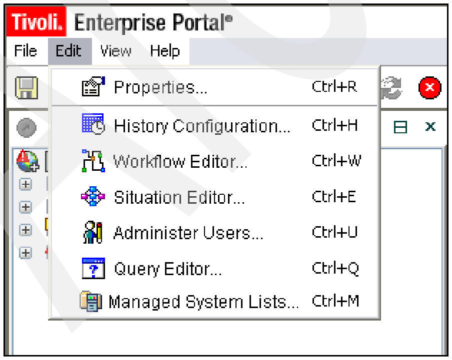


Figure 5-15 Edit Manage System Lists

The CICS_Production Manage System List (see Figure 5-16) was initially copied (using Create Another) from the *CICS MSL and then modified using the Add and Remove arrows to include only the appropriate CICS regions. The Available Managed Systems window contains the other currently known managed systems that could be added to this MSL, defined to other MSLs, or both.

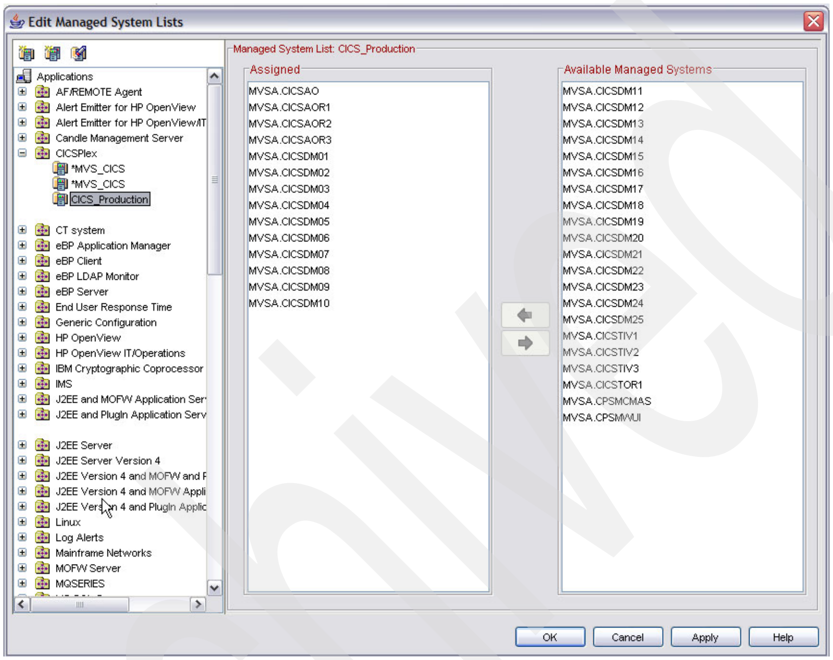


Figure 5-16 CICS_Production Manage System List

Similar to use of the RKC2XMnn DD DUMMY statement for OMEGAMON II for CICS to indicate which of multiple KOCCI collector address spaces in one LPAR will manage a CICS region, IBM Tivoli OMEGAMON XE for CICS on z/OS uses the RKCPXMnn DD statement to indicate which Tivoli Enterprise Monitoring Agent started task will manage a CICS region. You must add the DD statement into the CICS region JCL that you want to be redirected with a suffix other than 00. The Tivoli Enterprise Monitoring Agent started task should also include the same DD statement to indicate the connection.

Situation distribution

There are more than 200 product-provided situations (PPSs) for IBM Tivoli OMEGAMON XE for CICS on z/OS, all of which are distributed to the *CICS managed system list. Even though all the situations are not all set to autostart, they are distributed to the agent for use by the CICS regions in the event they are started. This can increase the startup time of the hub Tivoli Enterprise Monitoring Server and consumes additional CPU.

Review the situations to determine those you want to autostart and remove the *CICS from the distribution lists of any situations you do not want to start automatically or dynamically. IBM Tivoli distributes the situations so they will appear on the Tivoli Enterprise Portal Navigator tree associated with the correct Navigator Item (see Figure 5-17). However, many of the situations seen in Figure 5-17 are not autostarted and may not be required in your environment.

When you undistribute the situations, they no longer appear when you open the Situation editor to see the situations assigned to this managed system. The situations can still be viewed by selecting the Situation Editor.

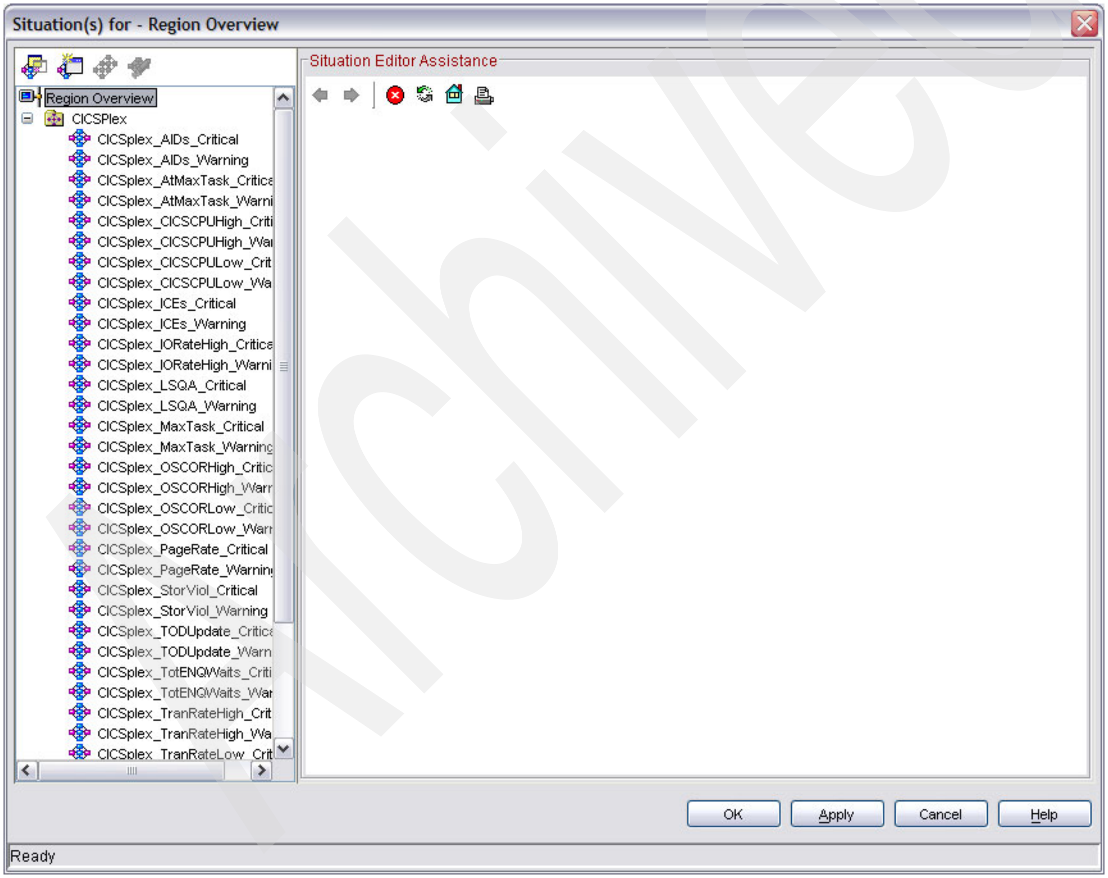


Figure 5-17 Situations associated with (Region Overview) object

Historical data collection

If your enterprise previously used the CandleNet Command Center (CCC) for CICS, you may have three large Persistent Data Store VSAM files to hold weeks of history information. With IBM Tivoli OMEGAMON XE historical data collection and the Tivoli Data Warehouse, you may not need to keep these VSAM files at the original size.

The Persistent Data Stores files need hold only 24 hours' worth of data at most. This reduces Tivoli Enterprise Monitoring Server and Tivoli Enterprise Monitoring Agent startup time, as it reads the records from all three PDS files before initialization is complete. The larger your history files, the longer it takes for Tivoli Enterprise Monitoring Server or Tivoli Enterprise Monitoring Agent to start.

5.5.2 IBM Tivoli OMEGAMON XE for z/OS

For some enterprises, the OS is the most closely watched component of performance and availability. It is often so closely watched that the same image is monitored multiple times with the same or similar thresholds. This is a waste of resources that can often be eliminated with the IBM Tivoli OMEGAMON XE architecture. The main focus of this section is how to avoid duplicate data collection using techniques unique to IBM Tivoli OMEGAMON XE for z/OS.

OMEGAMON II address spaces considerations

The following address spaces are required for IBM Tivoli OMEGAMON XE on z/OS processing:

- OMEGAMON for z/OS data collector, which executes within a z/OS-based Tivoli Enterprise Monitoring Server address space
- OMEGAMON II for MVS component CSA address space for common storage alerts as well as detailed displays in the classic interface

These components are optional; you may use them if you need their functions:

- OMEGAMON II for MVS component collector address space (RCOL) is not *required* (per se) by OMEGAMON for z/OS, but this address space is always executing as it provides deep-dive z/OS information.
- OMEGAMON II for MVS component Epilog historical data collector address space if history is desired.
- OMEGAMON II for MVS component ETE (End-to-End) address space if SNA network or TSO Host response by user ID is required. You may be able to quiesce the End-to-End address space (which can be shared with the OMEGAMON II for IMS, OMEGAMON II for CICS, and OMEGAMON II for Mainframe Networks components) because the pure SNA LU2 3270 session is no longer widely used. Most installations are using TN3270 sessions, which

make the response time information in ETE less valuable except for TSO host response time by user ID.

- OMEGAMON II for MVS component Epilog Zoom address space if VTAM (versus TSO) access to Epilog classic reports is desired. Epilog Zoom is a VTAM interface to EPILOG reports. If you do not use it from the 3270 terminal view, you can also stop the address space.

Not all OMEGAMON II CUA or OMEGAMON classic functions are available in IBM Tivoli OMEGAMON XE on z/OS, but IBM Tivoli OMEGAMON XE on z/OS does contain all of the exceptions from OMEGAMON II and OMEGAMON classic. If you are not using the 3270 CUA sessions, there is no need for starting the following OMEGAMON II CUA component address spaces:

- OMEGAMON II for MVS component's CUA presentation is not required if the customer uses classic interface.
- OMEGAMON II for MVS component's HDI address space, the CUA to Epilog interface, is not usable if OMEGAMON II for MVS CUA is not running.

Avoiding duplicate background collection

These recommendations apply to duplicate background collection with situations:

- Duplicate WLM data collection

  The following is relevant for sites that have not yet upgraded to IBM Tivoli OMEGAMON XE for z/OS and are still executing IBM Tivoli OMEGAMON XE for Sysplex *and* IBM Tivoli OMEGAMON XE for OS/390.

  The WorkLoad Manager (WLM) collector is designed to be shared among OMEGAMON II for MVS components and its Epilog facility and OMEGAMON for OS/390. This occurs only when all of these components run in the same address space as the Tivoli Enterprise Monitoring Server. If you configure the OMEGAMON for OS/390 in a separate agent address space, it starts a second copy of this collector, which doubles the processing required for WLM data collection.

- Duplicate situation drive data collection

  IBM Tivoli OMEGAMON XE for z/OS has product-provided situations that enable you to match the available thresholds in the OMEGAMON II for MVS component. These situations are installed with a sampling interval of 15 minutes, autostart set to no, and no distribution. Setting the situation sampling interval to match the auto-refresh interval used in OMEGAMON II for MVS CUA causes more frequent data collection and results in additional processing. Changing all situations' autostart has a similar effect because not all thresholds are evaluated in the default profile for OMEGAMON II CUA.

- Duplicate situation samples

  To ensure consistent exception analysis intervals, OMEGAMON for z/OS situation intervals were set to match OMEGAMON II for MVS component's CUA light frequencies. In the case of the Operator alerts, this results in five unique frequencies and five executions of the operator alerts data collector. Changing the situation intervals and increasing the frequency of situation evaluation will actually reduce the frequency of data collection because the situation synchronization mechanism will combine data collection requests.

  The default auto-refresh interval is 60 seconds, so you can set the situation interval to 60 seconds for the lights listed below that are one-times the auto-refresh interval. For the remainder of the lights, specify an interval for the situations that is equal to or less than the CUA intervals. Setting lower intervals takes advantage of situation synchronization and only two situations will be created for the operator alert table instead of five. The Syslog exception should be lowered to a one instead of two-minute interval to synchronize it with GRS, WTO, and SMF. The remaining operator alerts that would have been three or more minutes should be set to three. Table 5-3 lists the original OMEGAMON II for MVS interval multipliers.

Table 5-3 OMEGAMON II for MVS interval multipliers

Interval multiplier	CUA status light
1 times	CPU, key tasks, WTO buffers, SMF, GRS
2 times	DDR swap, Syslog
3 times	Batch, STC/APPC, TSO RTA, TSO host, Period 1 TSO, Workload, DASD, tape, CSA, OLTEP, HSM, GTF
4 times	Paging, storage, enqueue, RMF™
5 times	WTORS, Max. tasks
9 times	Channels, Key DASD

- Unnecessary situation samples

  IBM Tivoli OMEGAMON XE distributes situations that enable you to duplicate any of the exceptions done by the OMEGAMON II for MVS CUA interface. These situations are not autostarted. Exercise care when enabling these situations to prevent duplicate or unnecessary monitoring.

  For some binary conditions in OMEGAMON II for MVS, you should only define one situation for each, instead of defining both the Warning and Critical situations. These situations are:

  – Operator alerts for OLTEP and GTF STCs not being active; Syslog, RMF, or SMF not recording; DDR swap in progress; or break in the GRS ring.

– For DASD, alerts for dropped ready, not responding, no dynamic reconnect, and indexed VTOC lost.
– For paging, page dataset not operational alert.
– For tape, dropped ready and not responding alerts.

You do not have to enable some other situations, such as expanded storage exceptions, as there are no mechanisms to influence them in the current new zSeries processors. The expanded storage situations that should not be enabled are:

– OS390_CentraltoExpandedStor_Crit
– OS390_ExpandedOnlineFrames_Crit
– OS390_ExpandedToCentralStor_Crit
– OS390_Migration_Rate_Crit
– OS390_Migration_Rate_Warn
– OS390_Real_Stor_Migrate_Age_Crit
– OS390_Real_Stor_Migrate_Age_Warn

Duplicate on-demand data collection

On demand data collection is caused by end users, not by situations or background data collectors. This is normally one of the smaller sources of processing. The biggest sources of duplicate data collection are the existing OMEGAMON II for MVS sessions and dedicated mode sessions.

- Temporary duplicate overhead

  While you are migrating to IBM Tivoli OMEGAMON XE for z/OS, you may also be running the OMEGAMON II for MVS sessions in auto-update mode. This may be through OMEGAVIEW or directly by users. You need to create a migration plan to minimize overhead during conversion and create a timeline to eliminate previous exception analysis.

  IBM Tivoli OMEGAMON XE for z/OS can provide instantaneous access to the raw data that created the alert and is an excellent substitute for the numerous auto-update sessions many customers have to the same system.

- On-demand sampling

  On-demand overhead occurs when a user requests data. Every time a user opens a workspace, the Tivoli Enterprise Monitoring Server requests a data sample as specified in each view's query. Every time that workspace is refreshed automatically or when the user clicks Refresh, the server requests another sample. Workspaces with many data rows being returned and workspaces with automatic refresh can increase processing requirements.

- Situation distribution

  There are numerous predefined situations, none of which are distributed to managed systems lists. Distribute only the situations that you are going to autostart or plan to enable manually. If you distribute all of the situations, they will be propagated to the agents at Tivoli Enterprise Monitoring Server startup. This may simplify any subsequent activation procedures, but increases startup time. Review the situations to determine which ones you plan to use and add distribution lists only for those situations. After the situations are distributed, their alerts will appear on the Navigator items with which they are associated.

- Historical data collection

  With the former CandleNet Command Center for Sysplex, sites tended to allocate large sequential files known as PDS (Persistent Data Store) files to hold weeks of history. With IBM Tivoli OMEGAMON XE and its historical reports that are easily accessible from workspaces, the data-warehousing feature makes more sense. Customers with IBM Tivoli OMEGAMON DE for z/OS can also use powerful ODBC custom queries against the data warehouse repository. Besides having easier access to longer-term data, the migration of data older than 24 hours from the z/OS PDS enables you to reduce its size. This reduces the Tivoli Enterprise Monitoring Server startup time required to read in the records from all three PDS files before historical initialization was completed.

Shared DASD probe

The largest consumer of CPU in any Tivoli Enterprise Monitoring Server is often the sysplex Shared DASD probe. This is especially true of the Tivoli Enterprise Monitoring Server designated as the sysplex proxy. Unfortunately, this is often the hub Tivoli Enterprise Monitoring Server.

First, verify the amount of overhead associated with the Shared DASD probes to determine whether it is important in your environment to eliminate or reduce these cycles. The procedure to determine MVS Tivoli Enterprise Monitoring Server overhead is documented in .Examples using MVS INSPect. on page 71. In the hub Tivoli Enterprise Monitoring Server, you would expect to see high CPU usage in module KFACOM because the UADVISOR situation consolidates all of the data from all of the remote Tivoli Enterprise Monitoring Servers in the sysplex and sort-merges the data. You should also expect to see high CPU% in KDFDEVSU in all the Tivoli Enterprise Monitoring Server address spaces.

There are ways to migrate, reduce, or eliminate these CPU cycles:

- Controlling sysplex proxy overhead

  OMEGAMON for z/OS has background UADVISOR collectors that run on every LPAR where the product is configured. These collectors gather the data even if no predefined situations referencing these metrics have been activated. These situations control the data collection and the data merging into the sysplex proxy, a central location in a sysplex where data is evaluated by the product-provided situations. The process of sorting and merging this data causes additional CPU cycles to be consumed in the Tivoli Enterprise Monitoring Server responsible for this function.

The primary option to manage sysplex proxy resource consumption is to control the amount of DASD information being sent to the sysplex proxy for sort-merge processing. For example, 6,000 unit addresses on each of nine LPARs in a sysplex will require Tivoli Enterprise Monitoring Server to sort 54,000 rows of data. This may require a DASD Threshold Filtering situation to reduce the number of rows sent to the sysplex proxy.

You can determine the filtering impact of various criteria and simulate the effect of a DASD Filtering Threshold situation by using the OMEGAMON II for MVS component classic interface. From the DASD panel ZPATH (fastpath I.F), select thresholds for % busy (DUTnn) and response time (RSPnn), and review the number of devices returned. Keep in mind that these OMEGAMON II for MVS filters are joined by OR and your situation will be joined by AND, so it will return even fewer rows.

Important: Make sure that the DASD filtering situation interval is at least 10 minutes or data collection overhead may exceed sort CPU savings.

After you have a DASD Threshold Filtering situation specified and enabled, verify that the number of devices exceeding the situation thresholds is no more than 100 or some reasonable number. IBM Tivoli OMEGAMON XE for z/OS takes the superset of volumes that are returned from all managed LPARs in the sysplex and gathers metrics on those volumes from every LPAR. Properly set, there should be very little sort-merge processing from the IBM Tivoli OMEGAMON XE for z/OS product.

- Migrating sysplex proxy overhead

  You do not want the sysplex proxy to run in the hub Tivoli Enterprise Monitoring Server, which is busy performing many functions for the Tivoli Enterprise Portal Server, the remote Tivoli Enterprise Monitoring Servers, and potential locally attached agents. But the first Tivoli Enterprise Monitoring Server defined to configuration tool is the hub Tivoli Enterprise Monitoring Server and, by default, the configuration tool assigns the first Tivoli Enterprise Monitoring Server as the sysplex proxy. So configure the hub Tivoli Enterprise Monitoring Server to not be eligible for the sysplex proxy and this will offload cycles from the hub Tivoli Enterprise Monitoring Server. You can change this using the configuration tool after the first remote Tivoli Enterprise Monitoring Server has been defined to the hub Tivoli Enterprise Monitoring Server.

- Reusing or recollecting data

  The IBM Tivoli OMEGAMON XE for z/OS background data collector gathers Sysplex DASD information regularly, whereas the LPAR-specific component collects DASD data in response to sampling requests. When writing situations against DASD data, use the sysplex component rather than the LPAR component because the DASD data has already been collected. The same is true for any situations that access similar data such as enqueue metrics. You can get on-demand reporting in the Tivoli Enterprise Portal Server by viewing sysplex workspaces without re-collecting the data. If you want fresher samples or DASD that was eliminated by the DASD filter, use the LPAR component and use custom queries to reduce the volume of data.

- Disable shared DASD data mover

  If you have no interest in sysplex-wide shared DASD metrics, either reports or alerts, you might disable the shared DASD mover. Given the interaction between IBM Tivoli OMEGAMON XE for z/OS and IBM Tivoli OMEGAMON XE for Storage solutions as well as potential maintenance impact, you should contact the IBM Support Center if you want to pursue this option.

5.5.3 IBM Tivoli OMEGAMON DE on z/OS

This section presents specific tuning information for IBM Tivoli OMEGAMON DE on z/OS.

- Use the Session_Name and Status_Name

  Enterprise OMEGAVIEW, which is part of the IBM Tivoli OMEGAMON DE on z/OS entitlement, has logic on top of the ITMS:Engine (formerly referenced as CT/Engine) framework. Enterprise OMEGAVIEW has indexed Session_Name and Status_Name fields, which means that using these fields in a situation predicate filters out unnecessary evaluation through the index before going to the agent. This dramatically reduces processing.

  You can also use wildcarding with VALUE on the three Enterprise OMEGAVIEW processed attribute names. The asterisk must be at the end; for example: OD*

  Situation predicate order is less important with Enterprise OMEGAVIEW because this Tivoli Enterprise Monitoring Agent finds any Session_Name, Status_Name, or State attributes and processes them first. It then sends the result set to the ITMS:Engine framework to evaluate the remaining predicates.

  Note that Enterprise OMEGAVIEW will not use the index if you use the SCAN or STR functions in the Session_Name and Status_Name predicates. For example, you should specify `STR Session_Name = 1,CICS` as VALUE Session_Name = CICS*

  If the situation has no indexed attributes, the OMEGAVIEW RKLVLOG will contain the message `KWOIR103I Unable to optimize Situation xyz`.

  Review these situations to see whether you can make them more efficient.

- Status items

  The product-provided situations for Enterprise OMEGAVIEW reference every Status_Item in the indexed database. They look for `STATUS_NAME=W*A*R*`. (Workload, Alerts, Resource), which passes all Critical and Warning state items. Make the status item name more specific.

  Do not process all session types if you are interested in only one type, such as z/OS. If you have reasonable naming standards, you can say:

  ```
  Status_Item.Session_Name=MVS*
  ```

  The Tivoli Enterprise Monitoring Agent ignores Status_Items with blank Managed System Names and does not forward them. If you still want to use OMEGAVIEW/3270 during a migration (from OMEGAMON II to OMEGAMON XE), then clear the Managed System Name so you do not send alerts across the SDM (Status Data Manager) bridge as shown in Figure 5-18.

```
KMVCC2                 Change Session Information           More: +

Session Name . . . . . . . . . . . . . MVSAAOR2  Managed System . .
Type . . . . . . . . . . . . . . . . : OM/II-CICS
Description  . . . . . . . . . . . . . OM II CICS CICSAOR2

OMEGAMON II for CICS VTAM applid . . . XE12C2P
OMEGAMON for CICS VTAM applid  . . . . XE12C2C
CICS region name . . . . . . . . . . . CICSAOR2
OMEGAMON II for CICS profile . . . .             Blank for default
User profile suffix  . . . . . . . .             Blank for default
Logical rows . . . . . . . . . . . . . 255       99 - 9999
Additional user data . . . . . . . .
Logon profile name . . . . . . . . . . $DEFAULT +
Session automatic startup  . . . . . . Yes  +    Yes/No
Session update interval  . . . . . . . 60        15 - 9999 seconds
Collector timeout interval . . . . . . 90        (variable)
Session automatic reconnect  . . . . . Yes  +    Yes/No
Automatic reconnect limit  . . . . . . 0         0 - 99
VTAM LOGMODE table entry name  . . . . SNX32704

 F1=Help  F4=Prompt  **=Bkwd  F8=Fwd  F12=Cancel
```

Figure 5-18   OMEGAVIEW Managed System Name

  If you were uncertain about the meaning of the Managed System Name when defining OMEGAVIEW sessions and did not let it equal the Session_Name, and also have duplicate managed systems, the managed systems will continually knock each other offline. The managed systems report in with the same name (or names), which drives Tivoli Enterprise Monitoring Server processing with node status table registration cycling between offline and online and incurring additional processing.

5.5.4 IBM Tivoli OMEGAMON XE for IMS on z/OS

This section discusses some techniques for managing IBM Tivoli OMEGAMON XE for IMS on z/OS.

Background data collectors

Background collection options were controlled by the start-up parameters in RKANPAR(KOIxxP00). This is still necessary and some of this information, such as RTA, is shared with the IBM Tivoli OMEGAMON XE reports. If you are running the EPILOG historical collector, you can save lots of CPU if you disable the collection of IMS DASD devices. This is done by coding NORESC(DEV) in RKANPAR(KEIOPTxx). If you have thousands of volumes, this could be more than 50% of the classic collector's cycles.

Controlling auto discovery

If you have no interest in proactive or reactive monitoring of certain regions, you can explicitly exclude them from being discovered by coding parameters in the RKANPAR member KIPCNFG. These can be customized using the IBM Configuration Tool. The parameters enable you to manage the discovery process through include or exclude wildcards. They are for the IMSIDs and IRLMIDs that you do not want to monitor, such as IMS(imsid) NONE and IRLM(irlmid) NONE.

Conversely, you can exclude all regions by default and only code the ones that you want monitored, for example IMS(*) NONE and IRLM(*) NONE in conjunction with one or more sets of IMS(imsid) ALL and IRLM(irlmid) ALL. Remember, if they are not discovered they will not appear on the Tivoli Enterprise Portal navigator tree. If for some reason you installed and configured the product before you were ready to use it, you could disable autodiscovery by commenting out the RKANCMD member KIPSTART member's parameter IPDC AUTODISCOVER.

Controlling background data collectors

Disabling autodiscovery of control regions prevents the regions from being displayed on the Tivoli Enterprise Portal navigator tree. For regions that must be available for reactive monitoring, you can selectively disable background collectors with the same RKANPAR member KIPCNFG as Example 5-11 on page 189 shows. You can disable or control the sampling interval for the various collectors on a region-by-region basis. If the interval is zero, this component will not execute. These parameters are also configured using IBM Configuration Tool.

Example 5-11 IMS KIPCNFG values

```
CONFLICT INTERVAL(30) IRLM(BRLM)
COUPFAC  INTERVAL(30) IRLM(BRLM)
MSC      INTERVAL(30) IMSID(I51B)
RTAGRP   INTERVAL(30) IMSID(I51B)
RTAITEM  INTERVAL(30) IMSID(I51B)
```

Collection can be disabled temporarily without recycling Tivoli Enterprise Monitoring Agent to aid in problem or overhead determination. The modify commands shown in Example 5-12 stop and restart all of the collectors. If you modify KIPCNFG, this also make the changes effective.

Example 5-12 Command to stop and start IMS collectors

```
F stcname,IPDC STOP IRLM(irlmname)
F stcname,IPDC STOP IMS(imsname)
F stcname,IPDC START IRLM(irlmname)
F stcname,IPDC START IMS(imsname)
```

Product-provided and customer situations

If any product-provided situations are distributed to *IMS, they will evaluate the previously collected data. If you have disabled the discovery of regions that you do not wish to monitor, then the product-provided and system-maintained Managed Systems List (MSL) *IMS can be used. If you want to reactively view systems and proactively monitor selected IMS, then you should create and maintain your own MSL with just the regions that you are interested in monitoring.

Do not start and distribute situations that you will not react to or maintain. When situations generate false alerts, they cause wasted manpower in investigation and loss of faith in the validity of the alerts. They also may cause additional overhead in data collection, and they certainly waste cycles in evaluation of criteria that are unimportant.

Additional consideration for Version 3.1.0

The new version has a different underlying collection methodology. Previously, if a region was discovered and the background collector had an interval, the data was only refreshed on that interval. So to reactively view a test system or other less important system, it had to be discovered. When it was discovered, it started background collectors on the region. You could have set the collection intervals to be much less frequent, but then when you viewed it on the Tivoli Enterprise Portal, you were looking at less current data.

In V3.1.0 the situations, if active, drive the data collection. These situations are potentially driving the same 79 attribute groups from V2.2.0, but now there are 78 additional fields in these groups. As stated earlier, when you are monitoring with all of the product-provided situations, the overhead is the same as in the previous version. You can have less overhead if you do not activate all of the situations for all of the control regions.

An additional refresh on the Tivoli Enterprise Portal also drives data collection. Therefore, the potential for less overhead is in discovering regions for reactive monitoring without enabling situations. Regions that need less proactive monitoring can have fewer active situations, which means less data collection and overhead. In V3.1.0, not using *IMS and using MSLs for production or test can reduce data collection overhead.

5.5.5 IBM Tivoli OMEGAMON XE for DB2 on z/OS

This section discusses the following topics on tuning IBM Tivoli OMEGAMON XE for DB2 on z/OS:

- Sampling intervals

  The probes that collect the data for the various report containers can be invoked less frequently. The interval is controlled by the KDPCNFG member of RKANPAR. The interval is the number of seconds between sampling intervals. If the interval value is zero, then this component will not execute.

- Disabling collectors

  Collection can be disabled temporarily without recycling the Tivoli Enterprise Monitoring Server to aid in problem or overhead determination. The modify commands in Example 5-13 stop and restart all the collectors. These commands also make the changes in KDPCNFG effective.

Example 5-13 Commands to stop DB2 collector

```
F stcname,DPDC STOP IRLM(BRLM)
F stcname,DPDC STOP DB2(D411)
F stcname,DPDC START IRLM(BRLM)
F stcname,DPDC START DB2(D411)
```

- DB2 object analysis overhead

  If you have a lot of DB2 objects and Object analysis is enabled in the OMEGAMON II for DB2 component, you may see high overhead in KFACOM. This module creates indexes in core databases created from data that is moved from the remote Tivoli Enterprise Monitoring Servers to the hub Tivoli Enterprise Monitoring Server using XCF. You may also see high storage usage in extended storage subpool size (145-160). Use the SQL code similar to Example 5-8 on page 157 to disable UADVISOR_KDP_VLOBJECTS. Just change all of its references to the DASD situation to the one listed above.

6

Working with Tivoli Enterprise Portal

This chapter discusses several scenarios to guide you in using the IBM Tivoli OMEGAMON XE family of products. The discussion ranges from customizing the Tivoli Enterprise Portal to solving performance problems using IBM Tivoli OMEGAMON XE. The discussion in this chapter consists of:

6.1 Monitoring your z/OS environment

As was discussed in Chapter 2, “IBM Tivoli OMEGAMON XE components and architecture” on page 19, the primary interface for IBM Tivoli OMEGAMON XE and OMEGAMON DE is the Tivoli Enterprise Portal. IBM Tivoli OMEGAMON XE products running on z/OS maintain the 3270-based interface for various OMEGAMON classic and OMEGAMON II CUA products. These interfaces can be accessed directly using the 3270 terminal or emulation or through the portal Host on demand interface.

The discussion starts with a general walkthrough the Tivoli Enterprise Portal interface including out-of-the box navigation and other facilities that are available in Tivoli Enterprise Portal in 6.2, “Working with IBM Tivoli OMEGAMON XE” on page 193.

Tivoli Enterprise Portal provides a great deal of flexibility for operators to define new workspaces and customize existing ones. You can perform advanced functions for customizing the workspace, including working with background images and linking workspaces. This discussion is presented in 6.3, “Workspaces and links” on page 196.

Situations provide a mechanism for operators to get automatic notification of potential system problems. When a situation fires, the alert icon is shown in the resources tree and alert operators, providing an excellent early warning system. We discuss situation manipulation in 6.4, “Working with situations for z/OS” on page 211.

Historical data collection into Tivoli Data Warehouse can provide insight into how the system performs and can be used as an information source for forecasting future system performance. Although IBM Tivoli OMEGAMON XE V3.1.0 has not fully exploited the facilities provided by Tivoli Data Warehouse, such as summarization and pruning, the data collection facility is still readily available. We discuss this in 6.5, “Historical data collection” on page 221.

IBM Tivoli OMEGAMON XE on z/OS provides the basic monitoring for your z/OS system. We discuss some scenarios for z/OS-based monitoring, especially the new features that are available with IBM Tivoli OMEGAMON XE on z/OS V3.1.0, in 6.6, “IBM Tivoli OMEGAMON XE on z/OS new features” on page 226. Other z/OS-related scenarios with sysplex are discussed in 6.7, “Sysplex-related monitoring” on page 230.

We also provide some CICS-based scenarios for IBM Tivoli OMEGAMON XE for CICS on z/OS in 6.8, “CICS Short on Storage (SOS)” on page 240.

6.2 Working with IBM Tivoli OMEGAMON XE

This section discusses the general usage of IBM Tivoli OMEGAMON XE in Tivoli Enterprise Portal within our sample environment.

6.2.1 Typical monitoring scheme

Working with Tivoli Enterprise Portal relates to managing your environment's performance using both reactive and proactive mode.

- The reactive mode relates to working with alerts generated from running situations that potentially can cause performance problems. This is similar to the exception monitoring in OMEGAMON classic or keeping the light green for the OMEGAMON II environment.
- The proactive mode indicates that you go through workspaces to spot potential problems in your environment. This is greatly enhanced with Tivoli Enterprise Portal workspaces, which enable different types of charts to be displayed in a workspace to depict individual performance metrics.

Usage should focus on reactive mode for most operators as this has the least amount of overhead for the monitoring system. A carefully adjusted situation threshold provides an excellent source of exception-based monitoring for most common problems. Implementation of IBM Tivoli OMEGAMON XE should consider adjusting all running situations to tune their thresholds and execution intervals.

Problem determination for an impending performance may require traversing workspaces to spot potential bottlenecks that are not captured by any situation, hence not generating alerts. Experienced system programmers in performance areas should be using the workspaces to find any problems. When a problem is found and corrected, a situation should be created to check such indications automatically.

With this model, performance problem knowledge should be documented and checked automatically using situations. Impending problems can be detected early using automatic checking with situations. Documentation should facilitate the re-creation of situations in case of system changes and indicate the necessary action for operators, should the situation evaluate to true.

6.2.2 Physical navigator view

Working with workspaces to find performance metrics relates to navigating the tree view. The default navigator tree is the physical view. Figure 6-1 on page 194 shows our physical navigator tree.

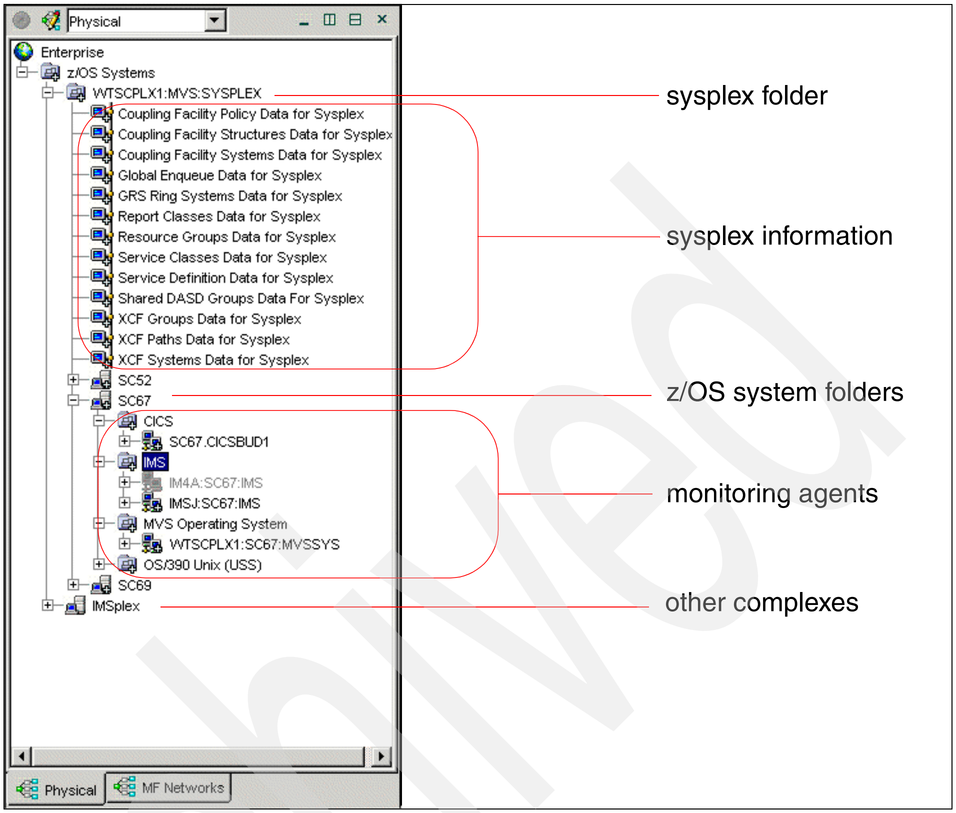


Figure 6-1 Physical navigator tree

The tree in Figure 6-1 has several groups of workspaces. Each node of the tree relates to a workspace.

- z/OS systems are grouped into sysplexes that are monitored in the enterprise.
- Sysplex-related workspaces that do not belong to any of the individual members are presented directly under the sysplex folder.
- Individual z/OS systems in the sysplex are defined as folders.
- The IBM Tivoli OMEGAMON XE products are shown as folders under the individual system on which the product is configured.
- Additional complex-based information such as IMSPlex is shown independently in the same level as the sysplexes.

6.2.3 Performance workspaces for z/OS

IBM Tivoli OMEGAMON XE provides a full range of workspaces that assist in performance analysis for z/OS systems. Figure 6-2 shows the Address Space Overview workspace.

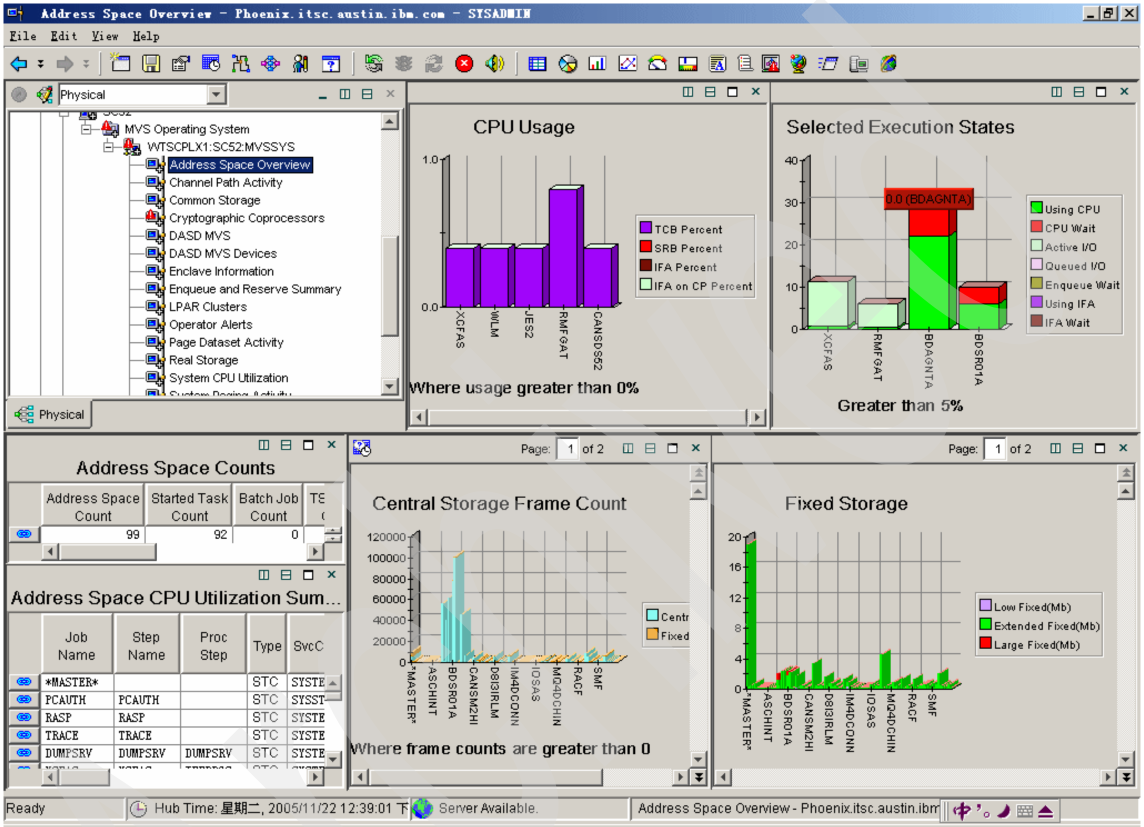


Figure 6-2 Address space overview

Figure 6-2 shows the main address spaces that consume CPU processing in the CPU Usage bar chart. The Selected Execution States bar chart depicts the execution states distribution of the main address spaces whose execution state (Using CPU, CPU wait, Active I/O, Queued I/O or Enqueued wait) occupies more than 5% of its life cycle.

The Address Space Counts table lists the statistics of address space count, started task count, batch job count, TSO user count, total enclave count, active enclave count, and inactive enclave count. Click the in the first column for further detail views of address space CPU utilization, storage, bottleneck summary, and common storage of active users. Click in the Address Space

CPU Utilization Summary table to list CPU usage summary and detail for each address space. The *Central Storage Frame Count* bar chart shows the central storage frame count allocated by the main address space. The Fixed Storage bar chart shows virtual storage area utilization measures for each address space.

6.3 Workspaces and links

The use of IBM Tivoli OMEGAMON XE solutions with the Tivoli Enterprise Portal user interface can entail reviewing information from multiple systems, subsystems, and components. This may require you to access information from multiple workspaces, which can be achieved with ease-of-use navigational aids provided by Tivoli Enterprise Portal, using the ability to link workspaces.

IBM Tivoli OMEGAMON XE has many product-provided links to assist with navigation among the product-provided workspaces. You can define more links between your custom workspaces and the product-provided workspaces.

- 6.3.1, “Linking overview” on page 196
- 6.3.2, “Link basics” on page 198
- 6.3.3, “Advanced links” on page 204

6.3.1 Linking overview

Links are saved as part of the workspace for which they are created. Links can span applications, logical views, and attribute groups. That is, you can link any workspaces regardless of their inherent relationship or apparent lack of relationship.

Tivoli Enterprise Portal provides two general types of links: simple and advanced.

- A simple link is a shortcut from one workspace to another that does not explicitly pass data from the source workspace to a target workspace. That is, simple links are like hyperlinks on Web pages that provide navigation from one page to another or from one Web site to another.
- An advanced link explicitly affects the data that is displayed on the target workspace. Advanced links pass filtering data to the target workspace. When executing an advance link, data from the link source is passed to the target workspace query or queries. The target workspace’s queries will filter the information and only return information that satisfies passed variables.

There is another aspect of defining a link. A link can be relative or absolute.

- Absolute link: Most user-defined links are absolute links. When you define an absolute link, you specify one specific workspace as the target workspace. Absolute links enable you to link to differnt levels in the Navigation tree.
- Relative link: Many product-provided links use relative links since host names are not known when the link is defined. When a relative link is executed, multiple target workspaces may match the target criteria. In this scenario, a menu is displayed from which the user selects the desired target. When a relative link links from a lower Navigator tree level to a higher level, there is no ambiguity and only one item satisfies the target criteria.

Figure 6-3 shows an example of links provided by IBM Tivoli OMEGAMON XE. Notice that all existing links, as well as the Link Wizard and Link Anchor facilities, are available from pop-ups. Also notice that when the link source is a tabular view, the link symbol is inserted to the left of the data rows.

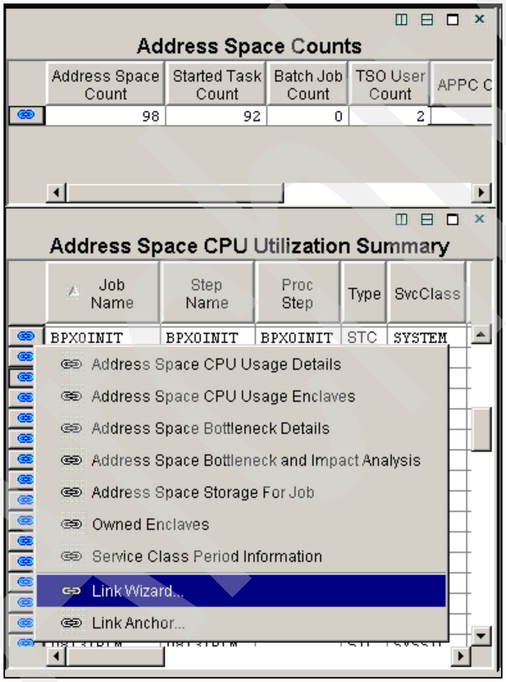


Figure 6-3 Links

6.3.2 Link basics

You can link from multiple areas within a workspace. You can link from:

- A navigator item (an item in the navigation tree).
- A table row in a tabular view. As shown in Figure 6-3, when a link exists in a tabular view, the link symbol is inserted in the far-left column.
- A graphic view item (an icon). In Figure 6-4, you can access the Link Wizard from an icon.

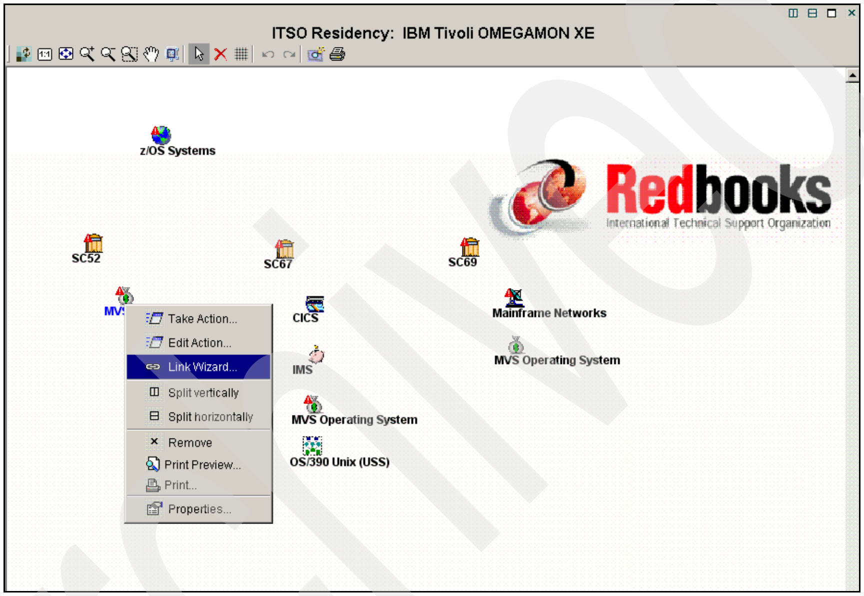


Figure 6-4 Linking from an icon in a graphic view

- The link symbol appears on the icon to indicate that a link exists. See the link on the MVS Operating System icon in Figure 6-5.

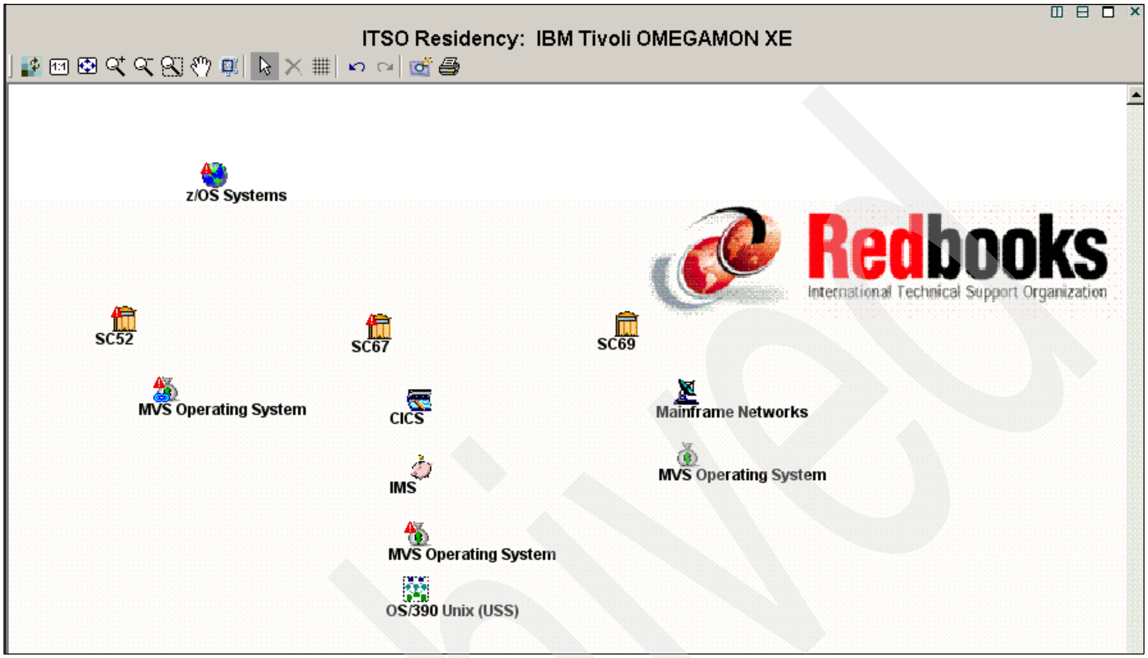


Figure 6-5 Graphic view with link from icon

► A bar chart or pie chart in a chart view. Figure 6-6 shows an example of using the bar chart view to create a link.

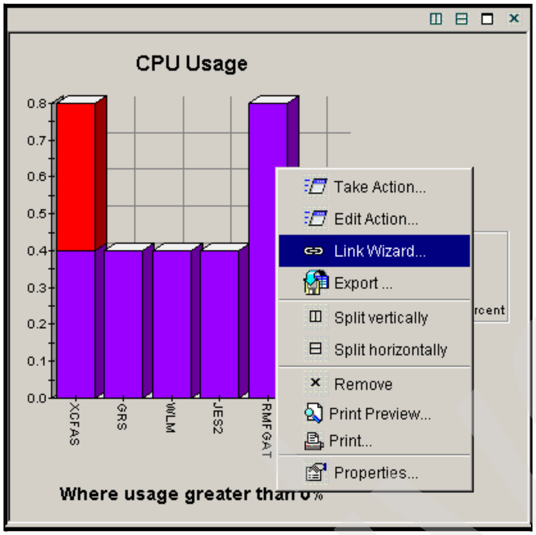


Figure 6-6 Linking from a (bar) chart

When the link source is a chart view, the link symbol is not provided. The Link To context menu in Figure 6-7 indicates that a link exists for the item.

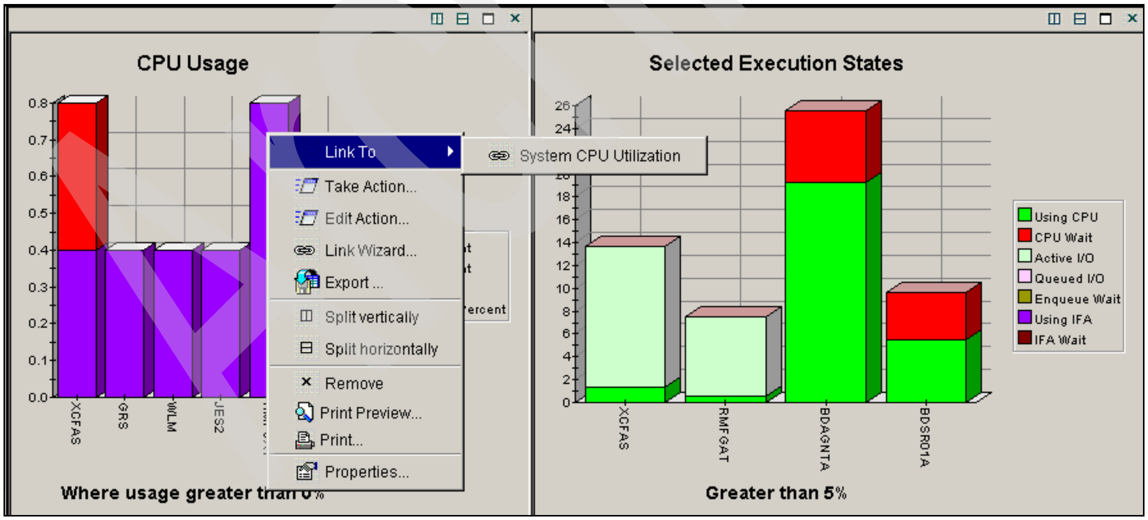


Figure 6-7 Link from CPU Usage bar chart to System CPU Utilization workspace

The following procedure builds a simple link for two workspaces:

1. Create or select your target workspace.
2. Access the source workspace and position the mouse cursor over the area in the source workspace where the link is required.
3. Right-click and launch **Link Wizard** from the context menu shown in Figure 6-3 on page 197.
4. Select **Define new link**. The initial page of the link wizard lists all existing links that could be selected and modified as required as shown in Figure 6-8.

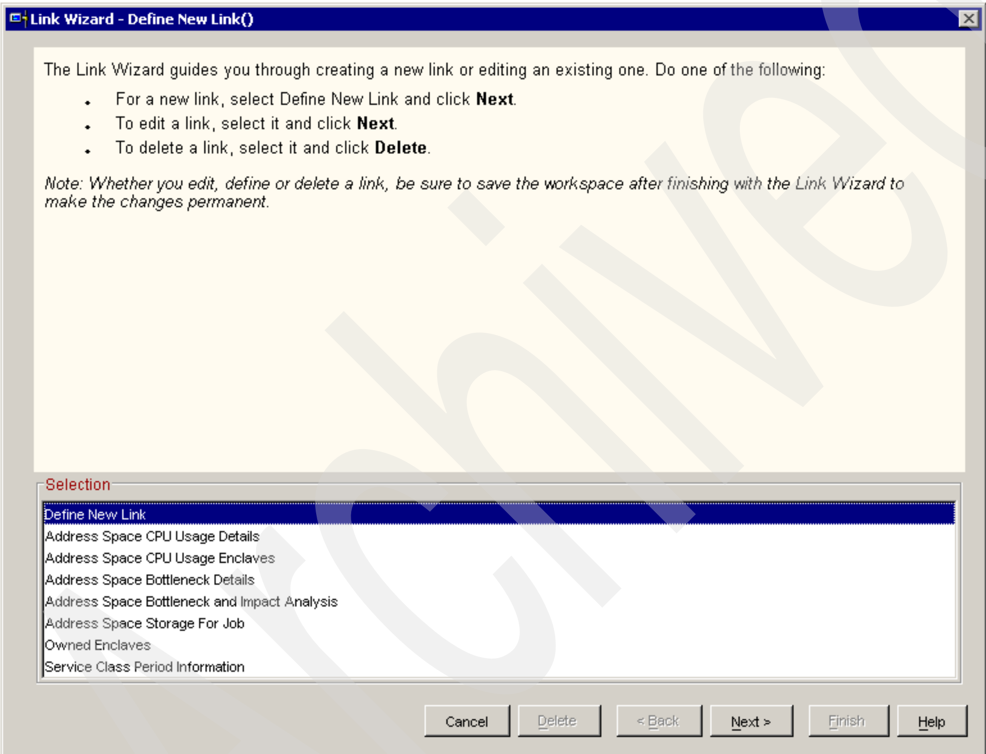


Figure 6-8 Define new link

5. Enter the link name and an optional description as shown in Figure 6-9. It is a good convention to name the link similar to the name of the target workspace to which you are linking.

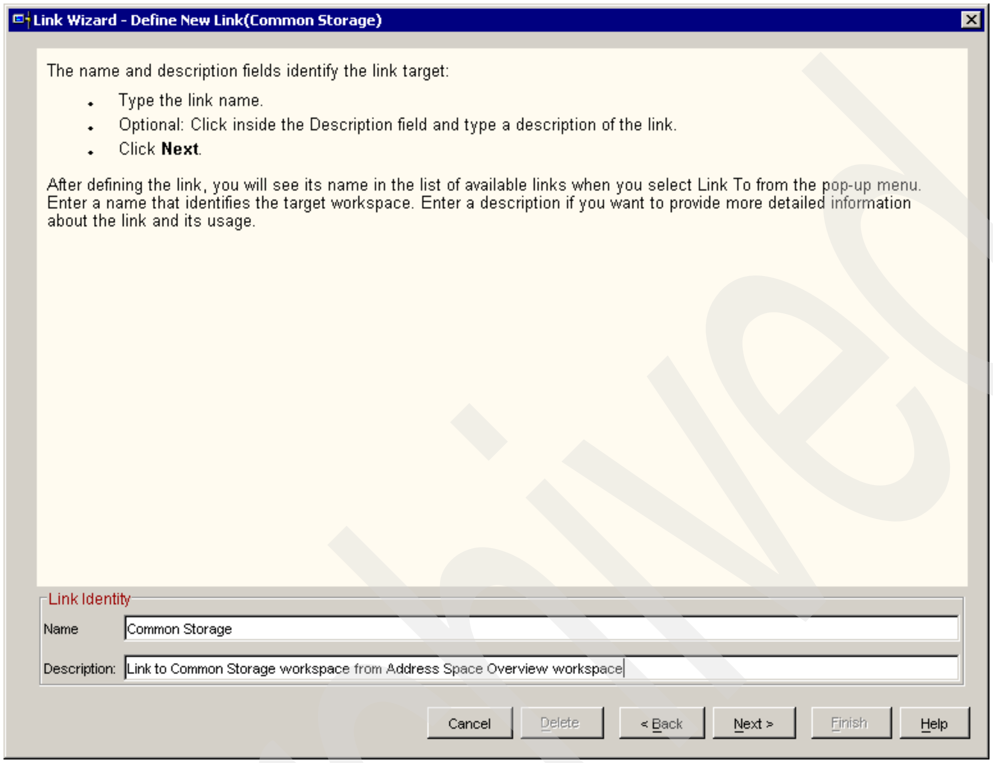


Figure 6-9 Specify link name and description

6. Select the target workspace from the dialog shown in Figure 6-10.
   a. Choose a navigator item to which to navigate.
   b. Select a workspace from the selected navigator item.
   c. Select **Absolute** mode.
   d. Click **Finish**.

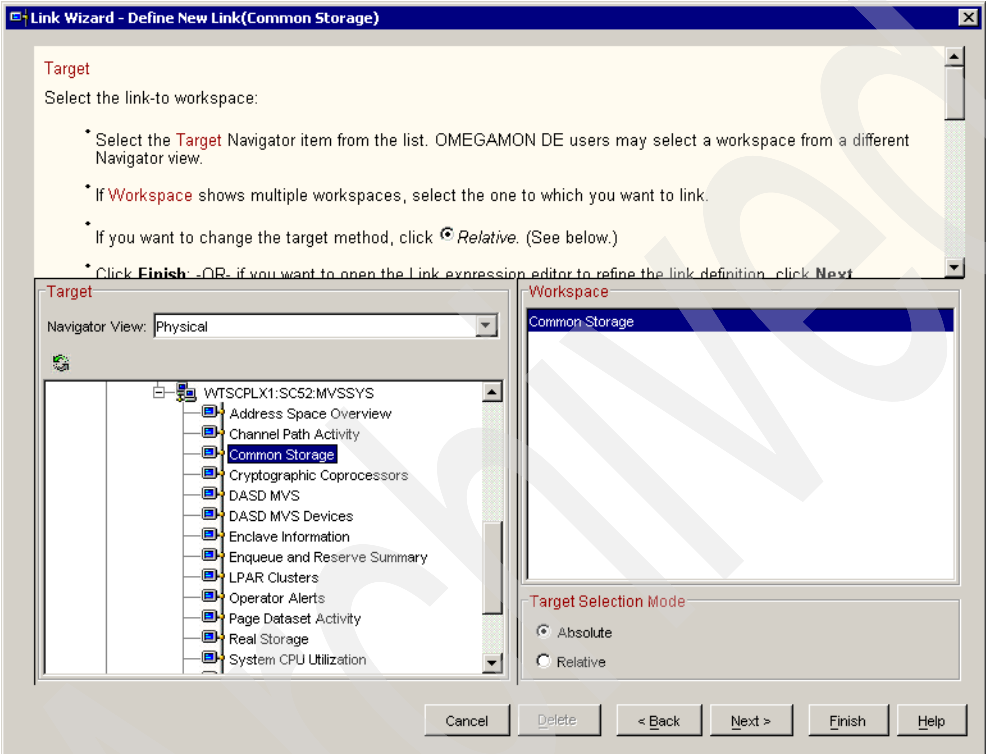


Figure 6-10 Specify target workspace

7. Execute the link to verify the results. Figure 6-11 shows the common storage workspace.

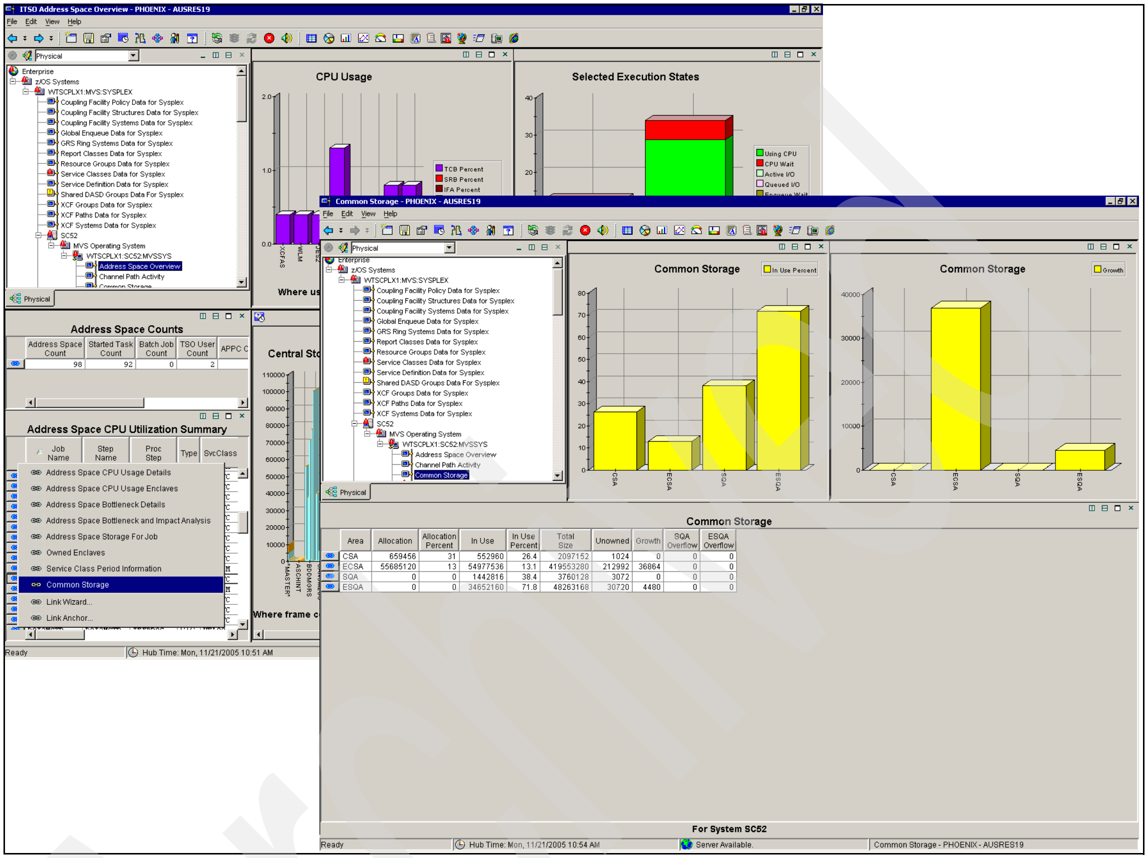


Figure 6-11   Link from Address Space Overview to Common Storage workspace

You can link any two workspaces even though there may not be a readily obvious relationship. One of the powerful features of links is that you can link any workspaces that you require.

6.3.3 Advanced links

Advanced links sometimes are called *filtered* links or *targeted* links. Advanced links provide the ability to pass values from the source workspace to the target workspace and dynamically filter the data that is displayed. Advanced links are defined similar to simple links using the Link Wizard. However, there is an extra step to complete the expression editor page.

For example, you want to see specific information related to each row of information in a tabular view. You may have hundreds of rows of information, so you would have to create simple links individually for each row to navigate to the appropriate detail data. This case requires you to define a dynamic link to pass filtering criteria.

The variables that you pass around are called link symbols. Link symbols are variables that get substituted with current Tivoli Enterprise Portal values at the time the link is executed. Link symbols are enclosed in $ signs.

The steps for creating your link using link symbols are:

1. Create the query (or queries) that you will use within your target workspace.
   a. Access the query editor from your toolbar by clicking the Query icon, or by pressing Ctrl+q on your keyboard.
   b. Create a new query or select an existing query and copy it to avoid disrupting the views of other users who might use this query in their workspaces.
   c. Include all of the new link symbols in your new query as shown in Figure 6-12.

> **Note:** Do not remove the $NODE$ link symbol, which is already listed in the query. It is used to pass the managed system name to the query. Without it, the query would not know what data source to access when it is issued.

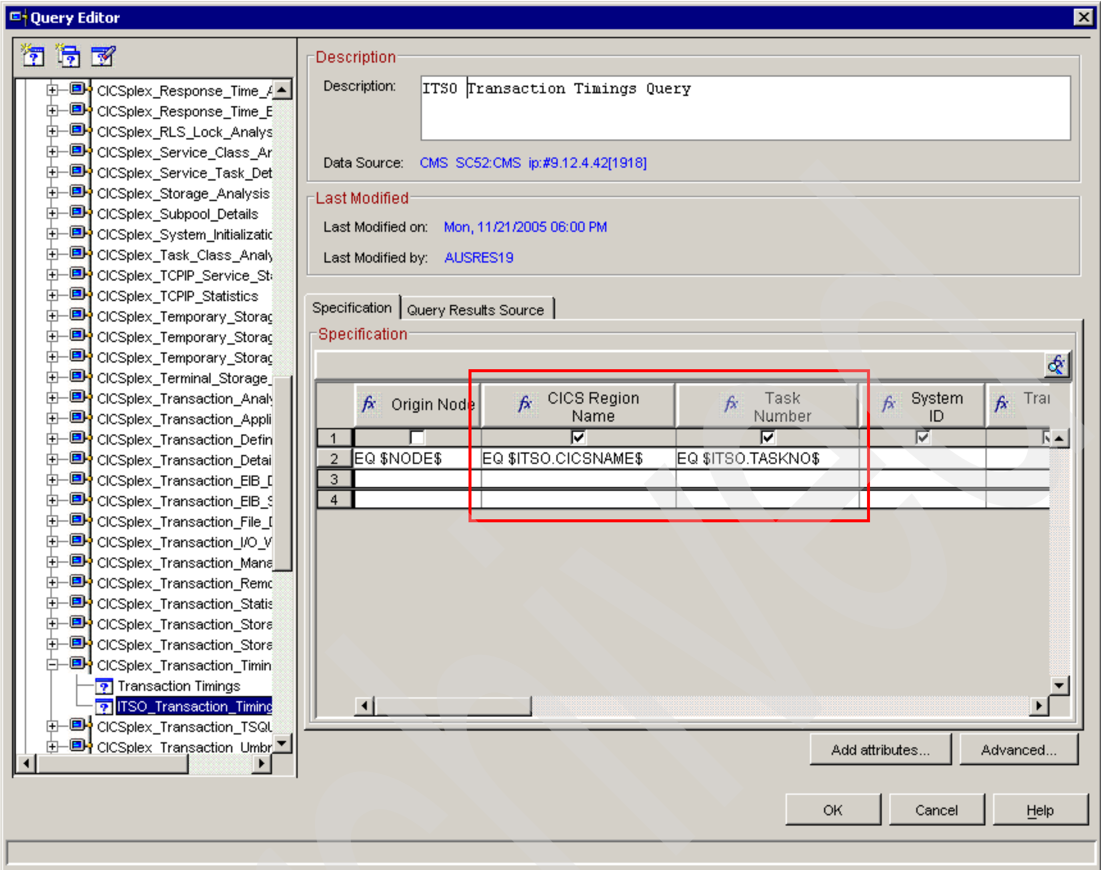


Figure 6-12 ITSO Transaction Timings query with link symbols

d. Test the expression.

2. Create the target workspace using the new queries that will provide the data.

   a. Create or access the target workspace.

   b. Modify the workspace to reference the appropriate query you defined in step 1 on page 205. Be sure the query contains the required link symbols.

Figure 6-13 shows the resulting view with multiple rows of information. When fully defined, this workspace will contain only one row of data. However, as the filtering criteria have not yet been passed to the query since the link is not yet defined, you see multiple rows.

Table

CICS Region Name	Task Number	System ID	Transaction ID	Elapsed Time	Dispatch Time	QR TCB Elapsed Time	Other TCBs Elapsed Time	CPU Time	Time in Suspend	Total Other Wait Times
CICSBUD1	00130	SC67	CEMT	00:07:43.284	00:00:00	00:00:00	00:00:00	00:00:00	00:07:43.283	00:00:00
CICSBUD1	00038	SC67	OSEC	2 Days	00:00:00.002	00:00:00.002	00:00:00	00:00:00.001	2 Days	00:00:00
CICSBUD1	00037	SC67	OSRV	2 Days	00:00:00.002	00:00:00.002	00:00:00	00:00:00.005	2 Days	00:00:00.027
CICSBUD1	00025	SC67	CSNE	2 Days	00:00:00.04	00:00:00.001	00:00:00.039	00:00:00.004	00:07:49.739	00:00:00.003
CICSBUD1	00023	SC67	CSHQ	2 Days	00:00:00.256	00:00:00.021	00:00:00.235	00:00:00.019	00:48:56.047	00:00:00.005
CICSBUD1	00021	SC67	CSNC	2 Days	00:00:00.032	00:00:00	00:00:00.032	00:00:00.001	2 Days	00:00:00.009
CICSBUD1	00020	SC67	CFQR	2 Days	00:00:00	00:00:00	00:00:00	00:00:00	2 Days	00:00:00.001
CICSBUD1	00019	SC67	CFQS	2 Days	00:00:00.023	00:00:00.014	00:00:00.009	00:00:00.014	00:01:11.929	00:00:00.007
CICSBUD1	00008	SC67	CSTP	2 Days	00:00:09.408	00:00:09.377	00:00:00.03	00:00:07.777	00:00:00.364	00:00:05.246
CICSBUD1	00006	SC67	CSSY	2 Days	00:00:00.007	00:00:00.007	00:00:00	00:00:00.006	00:48:58.736	00:00:00
CICSBUD1	00005	SC67	CSSY	2 Days	00:00:00.134	00:00:00.074	00:00:00.059	00:00:00.007	15:09:41.542	00:00:00
CICSBUD1	00003	SC67	CSOL	00:05:22.538	00:00:00	00:00:00	00:00:00	00:00:00	00:05:22.538	00:00:00

Figure 6-13 Query for target workspace (unfiltered)

3. Create a new link from the source workspace:

   a. Access the source workspace for your link and launch the Link Wizard on the appropriate link source.

   a. Name the link.

   b. Select the previously created target workspace.

   c. Instead of selecting Finish as you did when defining a Simple Link, click **Next**. This accesses the link expression editor, which enables you to assign values and expressions to your link symbol.

   d. Click the link symbol, which is displayed in the left pane of the expression editor.

   e. Assign the appropriate attribute to the Link Symbol. This populates the Expression field.

   > **Note:** It is easy to select the item twice. You can validate that by looking for two of the same expressions in the Expression field. If you accidentally added two expressions, remove one or clear the field, and select it again.

   f. Click **Test** to verify the expression.

   g. Click **Finish** in the link wizard to finalize the link.

h. Test the link by accessing the new link from one of the table rows in the Transaction Totals table.

4. Assign a value or expression to all link symbols that are displayed in the link wizard expression editor. See Figure 6-14, which shows the link symbol specifications for two variables: ITSO.CICSNAME and ITSO.TASKNO.

> **Note:** The expression editor enables you to build expressions for the parameters or link symbols to be passed to the target workspace. The expressions can be used by the target workspace queries or by the view display header. See “Header link symbols” on page 210.
>
> The expression window shows the specifications for the value that will be passed in the link symbol. These specifications were selected from the available attributes in the AllowableTerms tree.

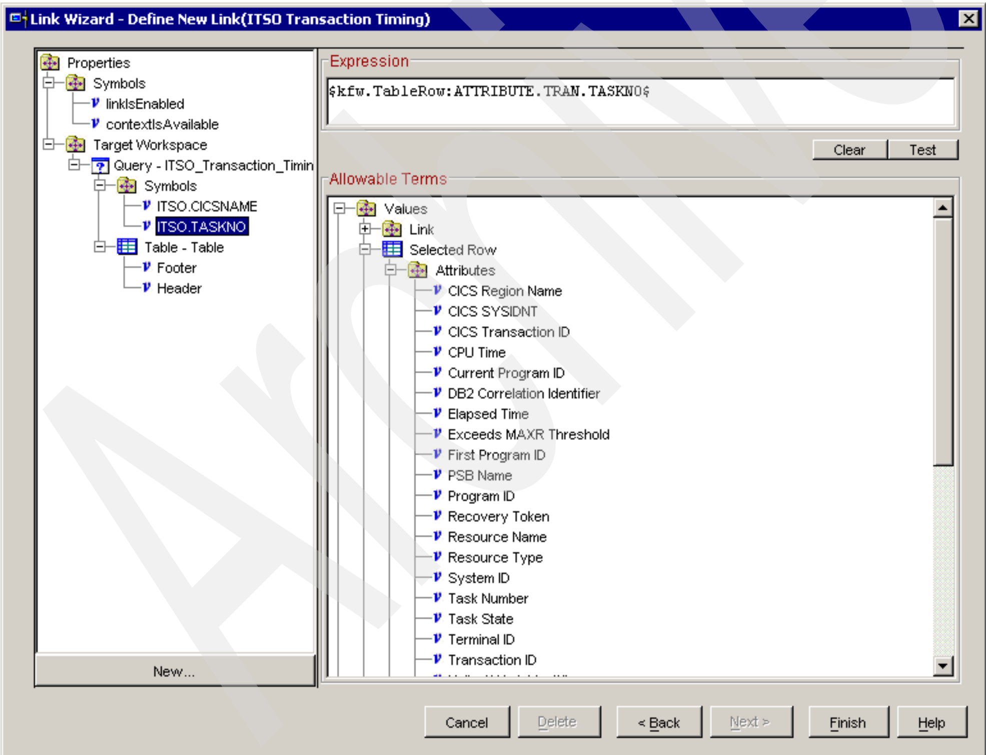


Figure 6-14 Link Symbol specifications for ITSO.CICSNAME and ITSO.TASKNO

5. Validate the link symbol resolution by clicking **Test**, which opens a dialog box similar to Figure 6-15. Verify this with the data from the target query in Figure 6-13 on page 207 to ensure valid format and value.

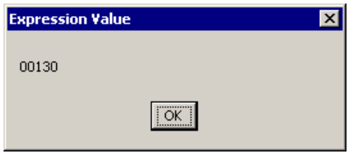


Figure 6-15 Test Link symbol resolution

6. Test the link and validate the target workspace contents. Figure 6-16 shows additional details for a single row of information, which is all that would satisfy the filtering criteria passed in the link symbol.

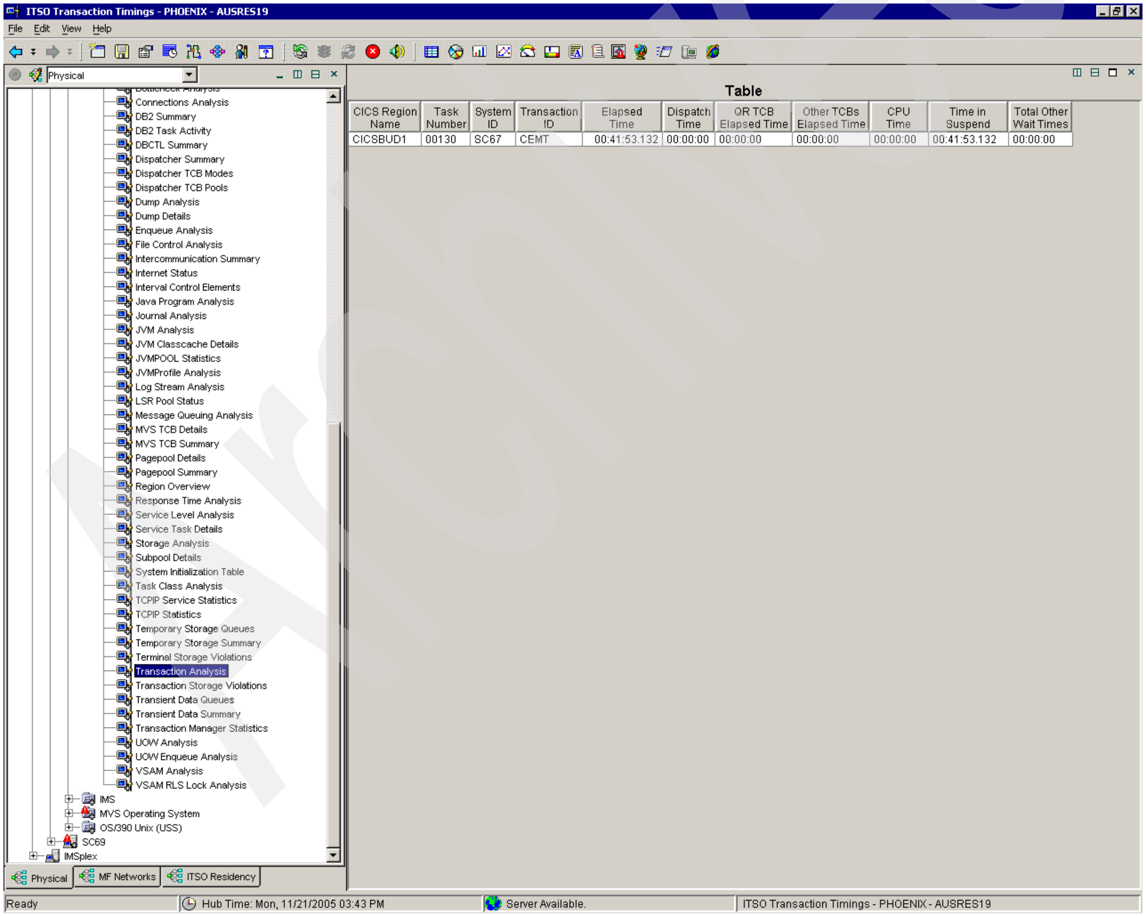


Figure 6-16 Advance Link target workspace with one row of data

6.3.4 Additional linking facilities

This section discusses some additional topics about linking workspaces.

Header link symbols

You can provide values from the source workspace to the target workspace for dynamically populating the header or footer in a view. The header or footer can contain a substitution for the link symbol that contains the passed value. Static text can be included in the expression by enclosing the text in double quotes. ("")

In the expression editor in Figure 6-17, you will find Header link symbols in the left-side tree: one for each embedded view. Select a Header link symbol and type the desired expression. Include a substitution using the $...$ value by selecting an allowable term. Text portions are concatenated using the plus (+) operator.

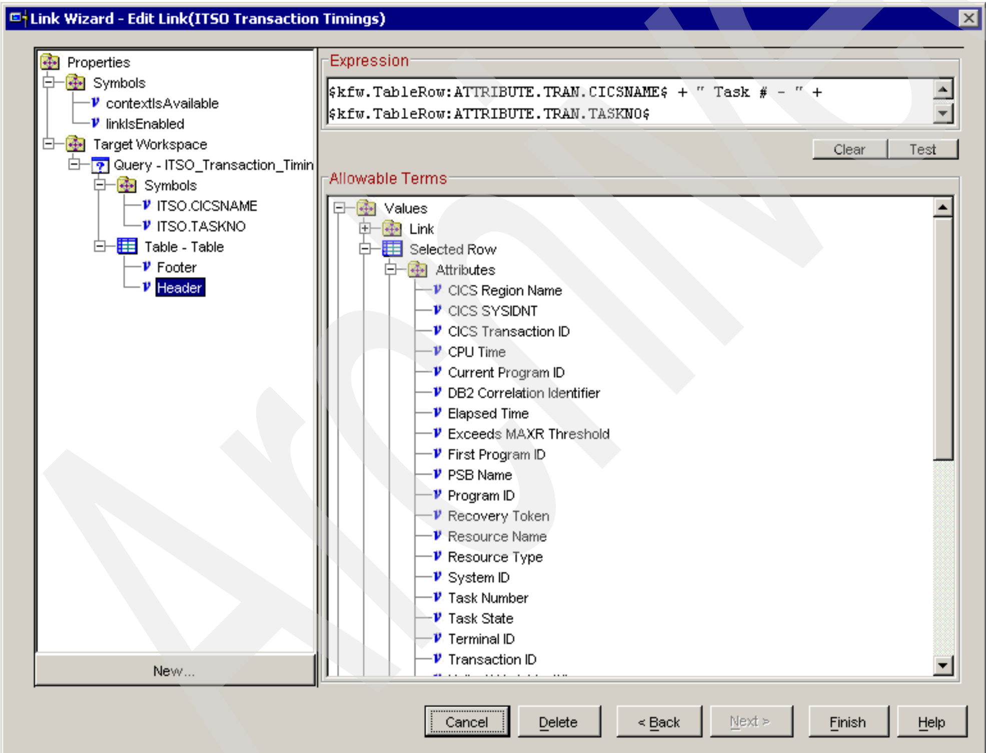


Figure 6-17 Header and footer link specifications

Figure 6-18 shows the resulting header and footer areas. The CICS Name, Task number, and Transaction ID are populated with information passed by the link.

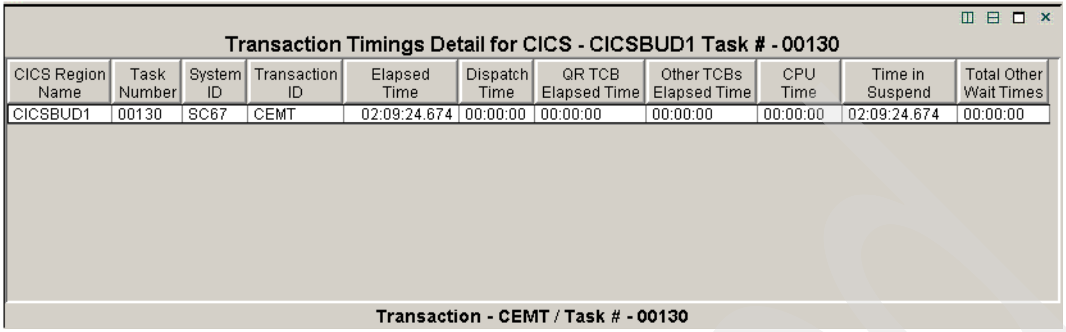


Figure 6-18 Header and footer link symbols

Linking across navigators

You can create links to workspaces in other navigators. When selecting the target in the Link Wizard, switch to another Navigator or Logical view and select the target workspace.

> **Note:** During execution, if a workspace with the same name (as the workspace in the targeted Navigator item) is available in the source Navigator, the link will switch to that workspace rather than the intended workspace in the different Logical view.

6.4 Working with situations for z/OS

This section illustrates how you can use situations for z/OS information. Situations alert you of potential system problems.

6.4.1 Predefined situations for z/OS

To alert of impending problems and to simplify analysis when problems occur, you can activate the predefined situations or create your own situations. Figure 6-19 on page 212 shows the predefined situation list for the z/OS system.You can access it by clicking the Situation editor button ( ) and expanding the MVS system tree.

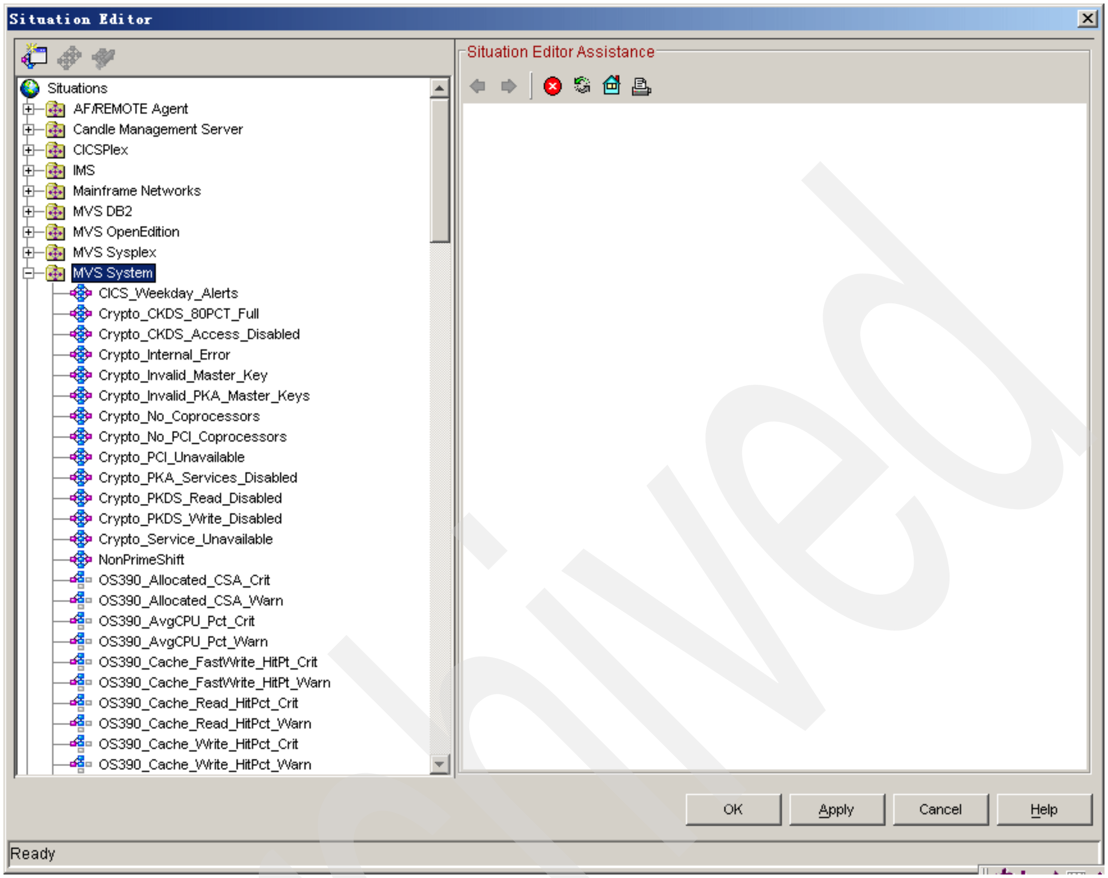


Figure 6-19 Predefined situations for MVS System

Most of the predefined situations shipped with OMEGAMON on z/OS are not automatically started. You must start these situations before they can monitor your system. You should examine the predefined situations and customize them with threshold values that match to your site's policies, goals, or monitoring requirements before you activate them. If you decide to make changes to a predefined situation, create a copy of the situation using the Create Another context menu and modify the copy. Otherwise, the changes you make will be overwritten the next time the product is updated or upgraded.

To activate or modify a situation, use the Situation editor of the Tivoli Enterprise Portal. You can then:

- Distribute or assign the situation to one of more managed systems or managed system lists.

- Start the situation: If you select Run at start up, the situation will start automatically when Tivoli Enterprise Monitoring Server starts.

6.4.2 Monitoring paging and virtual storage

IBM Tivoli OMEGAMON XE on z/OS provides a collection of predefined situations and attribute groups for you to use for monitoring paging and virtual storage on z/OS. Table 6-1 lists the predefined situations for paging and virtual storage. They are not automatically started. You must customize them before enabling them if you want to use them.

Table 6-1 Predefined situations for paging and virtual storage

Situations	Description
OS390_Allocated_CSA_Crit	Critical alert for allocated CSA storage more than 95% full
OS390_Allocated_CSA_Warn	Warning alert for allocated CSA storage more than 90% but less than 95% full
OS390_Common_PageDS_PctFull_Crit	Critical alert for common page dataset more than 95% full
OS390_Common_PageDS_PctFull_Warn	Warning alert for common page dataset more than 60% but less than 80% full
OS390_CSA_Growth_Crit	Critical alert for CSA growth in use during the last interval more than 50%
OS390_CSA_Growth_Warn	Warning alert for CSA growth in use during the last interval more than 35% but less than 50%
OS390_CSA_Growth_Crit	Critical alert for CSA growth in use during the last interval more than 50%
OS390_ECSA_Allocation_Pct_Crit	Critical alert for allocated ECSA storage more than 95% full
OS390_ECSA_Allocation_Pct_Warn	Warning alert for allocated CSA storage more than 90% and less than 95% full
OS390_Local_PageDS_Errors_Crit	Critical alert for the number of local page dataset errors more than 5
OS390_Local_PageDS_Errors_Warn	Warning alert for the number of local page dataset errors more than 1 but less than 5
OS390_Local_PageDS_PctFull_Crit	Critical alert for local page dataset more than 35% full

Situations	Description
OS390_Local_PageDS_PctFull_Warn	Warning alert for local page dataset more than 25% but less than 35% full
OS390_Page_Rate_Crit	Critical alert for page rate (pages per second read or written for this dataset) less than 0.0
OS390_Page_Rate_Warn	Warning alert for page rate (pages per second read or written for this dataset) less than 0.0
OS390_PageDSNotOperational_Crit	Critical alert for the number of page datasets that are not operational is more than 5
OS390_PageDSNotOperational_Warn	Warning alert for the number of page datasets that are not operational is more than 0 but less than 5
OS390_PLPA_pageDS_PctFull_Crit	Critical alert for PLPA page dataset less than 0
OS390_PLPA_pageDS_PctFull_Warn	Warning alert for PLPA page dataset less than 0
OS390_System_Page_Rate_Crit	Critical alert for system page rate (the rate at which paging occurs over all storage areas: common, private, and system) less than 0.0
OS390_System_Page_Rate_Warn	Warning alert for system page rate (the rate at which paging occurs over all storage areas: common, private, and system) less than 0.0
OS390_System_Pagefault_Rate_Crit	Critical alert for system page fault rate (the number of page faults per second) more than 1000000000
OS390_System_Pagefault_Rate_Crit	Warning alert for system page fault rate (the number of page faults per second) more than 1000000000

Table 6-2 lists the predefined attributes in the attribute group for paging and virtual storage. You can add the attribute in the Condition tab of situation editor to make the conditions of your created situations more specific to your monitoring requirement.

Table 6-2 Attribute group for paging and virtual storage use

Attribute group	Description
System paging activity	Provides information about factors taking place in your z/OS system that affect system paging rates, including the ASM queue length, datasets not operational, expanded storage pages moved, managed system, system page rate, page fault rate, and unreferenced interval count.
Common storage	Displays information about four important areas of common storage: common service area (CSA), extended CSA, system queue area (SQA), and extended SQA, including allocation amount, allocation percent, area, ESQA overflow amount, growth, in use amount, in use percent, managed system, SQA overflow, total size, unowned amount
Page dataset activity	Provides information about availability and response time for specific page dataset, including address, dataset type, errors, managed system, page rate, percent full, response time, volume

The following procedure creates a situation to monitor common storage:

1. Navigate to the Common Storage workspace for SC52. In the workspace, access the Situation editor by right-clicking the **Common Storage** navigator item and selecting **Situations** as shown in Figure 6-20.

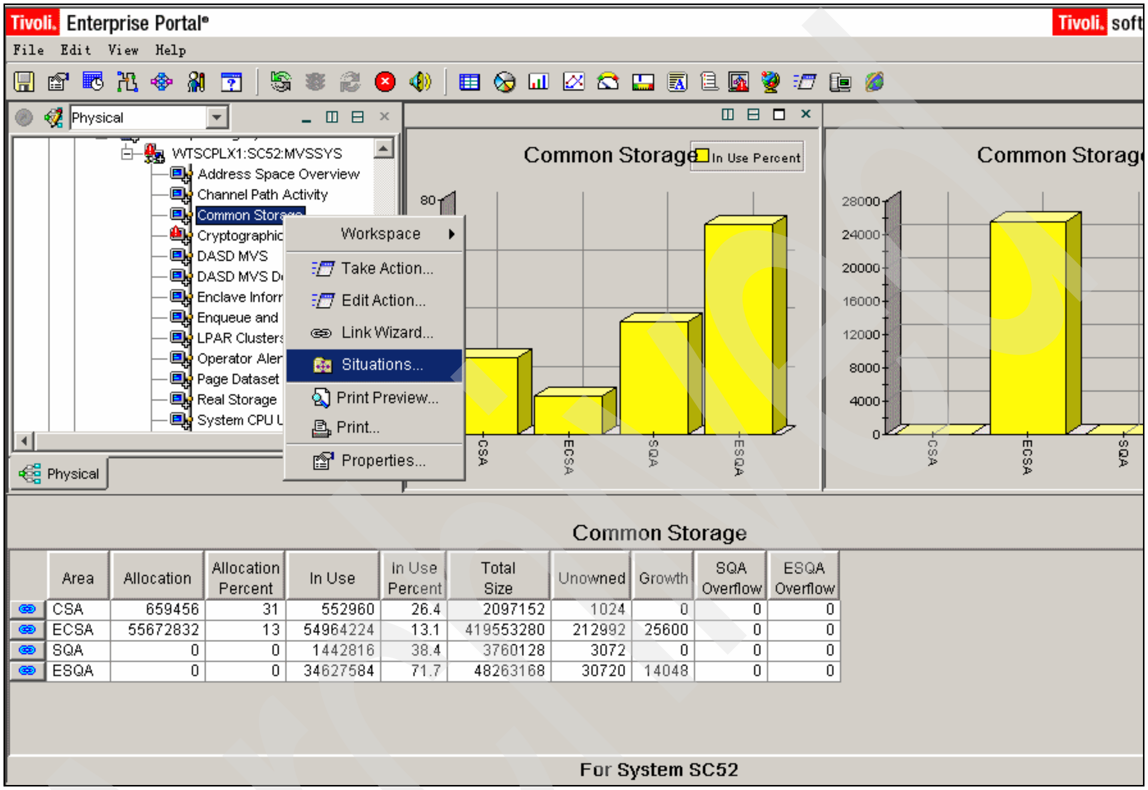


Figure 6-20 Create situation to monitor common storage

2. In the Situation editor, creates a new situation. Type a name and description for the situation, for example, `Common_storage_monitor_warn` as shown in Figure 6-21.

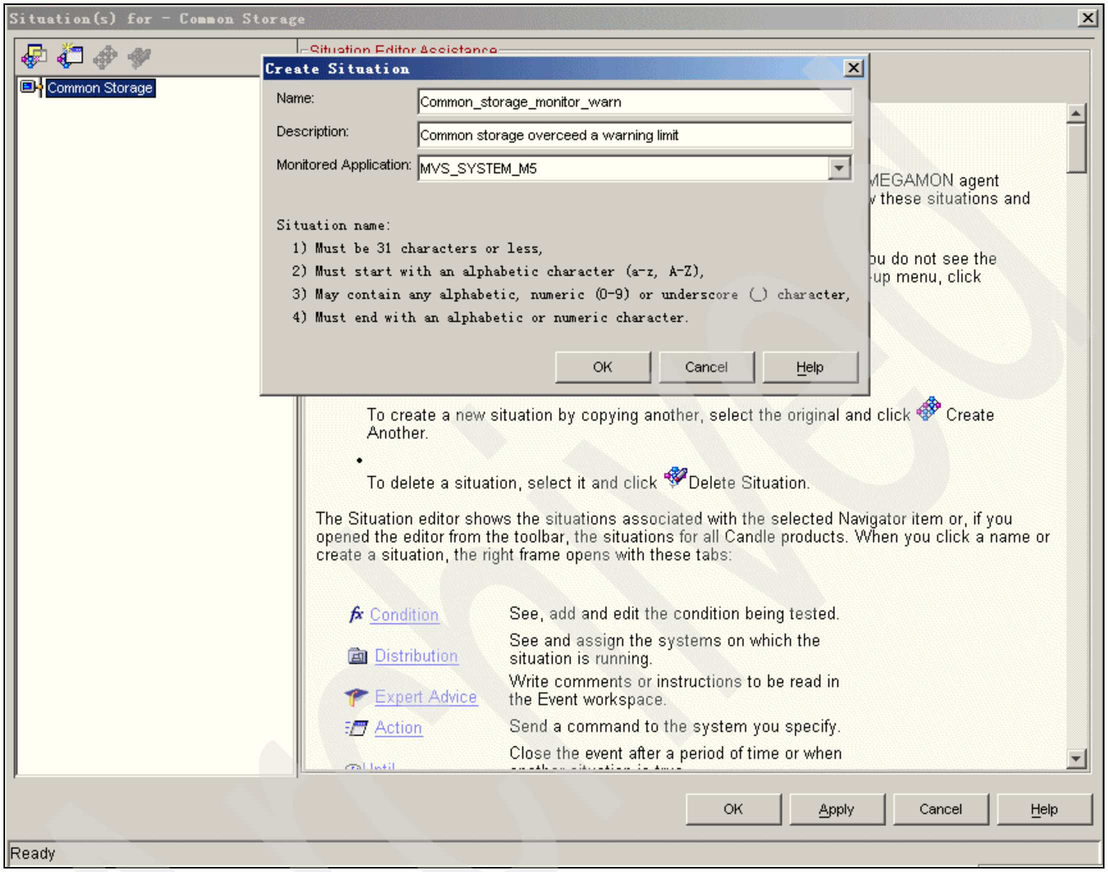


Figure 6-21 Create situation to monitor common storage - new situation

3. The new situation requires an attribute. In the Select Attribute dialog, select the **Common Storage** attribute group and the **Allocation Percent** attribute as shown in Figure 6-22. Click **OK**.

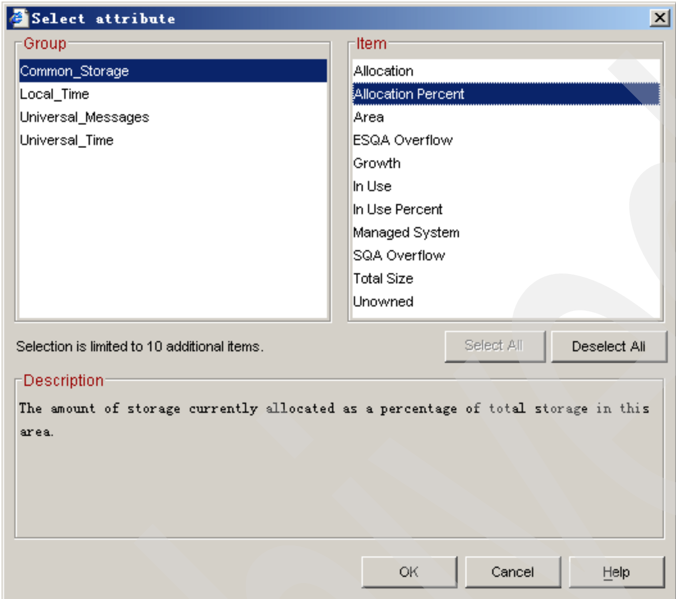


Figure 6-22 Create situation to monitor common storage - selecting attribute

4. After selecting one attribute, Figure 6-23 shows your situation. Click **Add attributes** to add new attributes that we want. Using the Select Attribute window, we selected **SQL overflow** and **ESQL overflow** attributes.

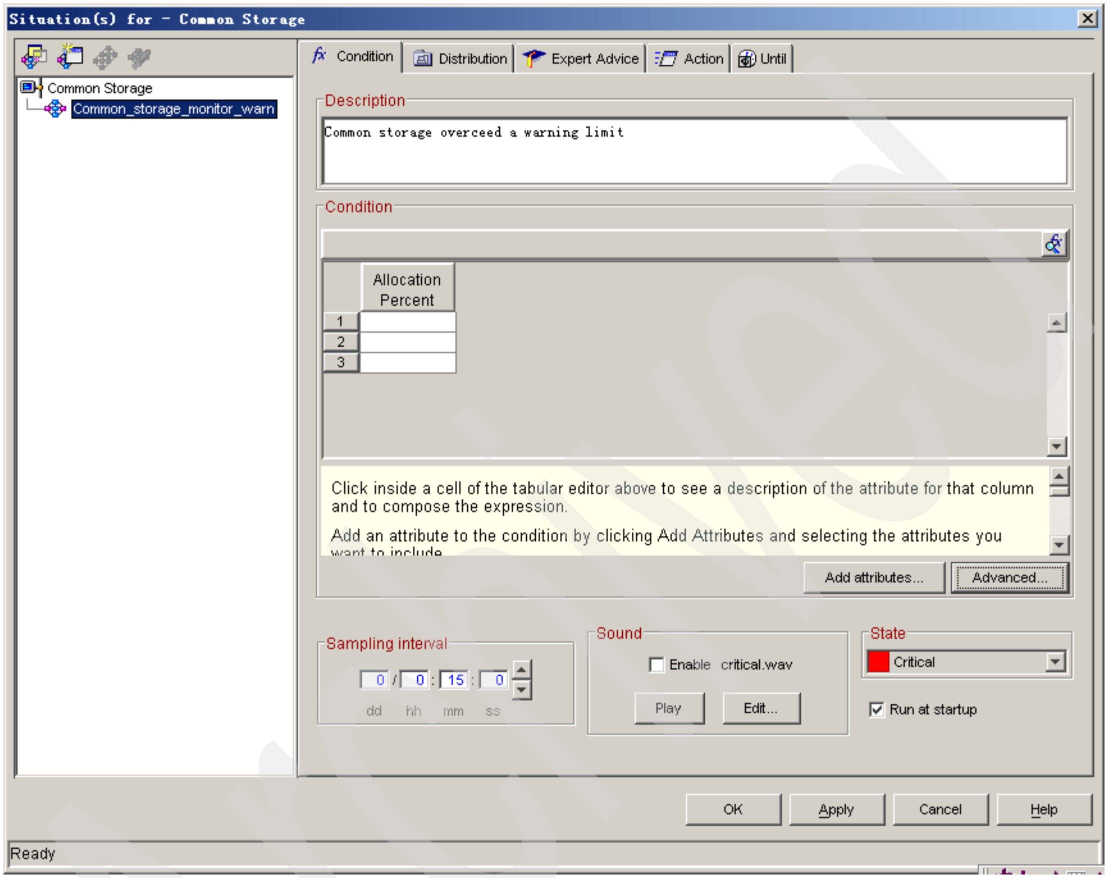


Figure 6-23 Create situation to monitor common storage - situation dialog

5. Modify the thresholds or conditions that the situation will apply, as shown in Figure 6-24. You can also modify the sampling interval of your new situation. Having a condition listed in multiple rows as shown means that any condition will trigger the situation (OR logic).

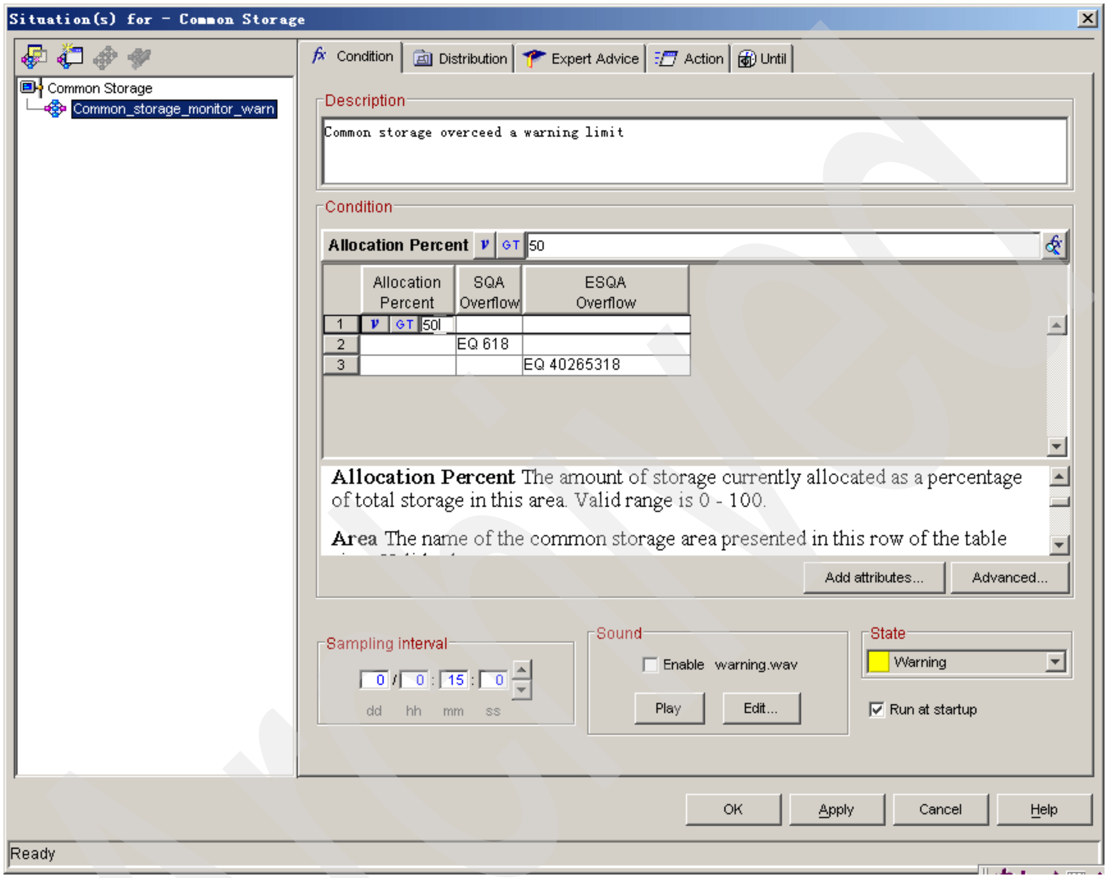


Figure 6-24 Create situation to monitor common storage - condition and sampling

6.5 Historical data collection

The Historical Data Collection configuration program, invoked from Tivoli Enterprise Portal, enables you to configure the historical data collection.

Configuration is done by attribute group basis. You can configure collection for different attribute groups at different intervals. However, for a given attribute group, you must apply the same historical collection options on all Tivoli Enterprise Monitoring Servers. Not all attribute groups can collect historical data, because collecting historical data for these attribute groups is not appropriate or would have a detrimental effect on performance. Only those attribute groups for which data can be collected are listed in the Configuration dialog.

Regardless of what location you specify for data collection, all data for a sysplex is collected at the sysplex proxy. Because the identity of the sysplex proxy can change (for example, when one system is taken offline and the backup becomes the proxy), you must configure historical data collection on all Tivoli Enterprise Monitoring Servers that are eligible as backups for the sysplex proxy and start collection on those servers.

Figure 6-25 on page 222 shows the historical configuration panel that you can access from Tivoli Enterprise Portal by clicking the Historical Data Configuration icon () from the tool bar. The Collection Status window shows that we have not defined a historical collection for IBM Tivoli OMEGAMON XE on z/OS so far.

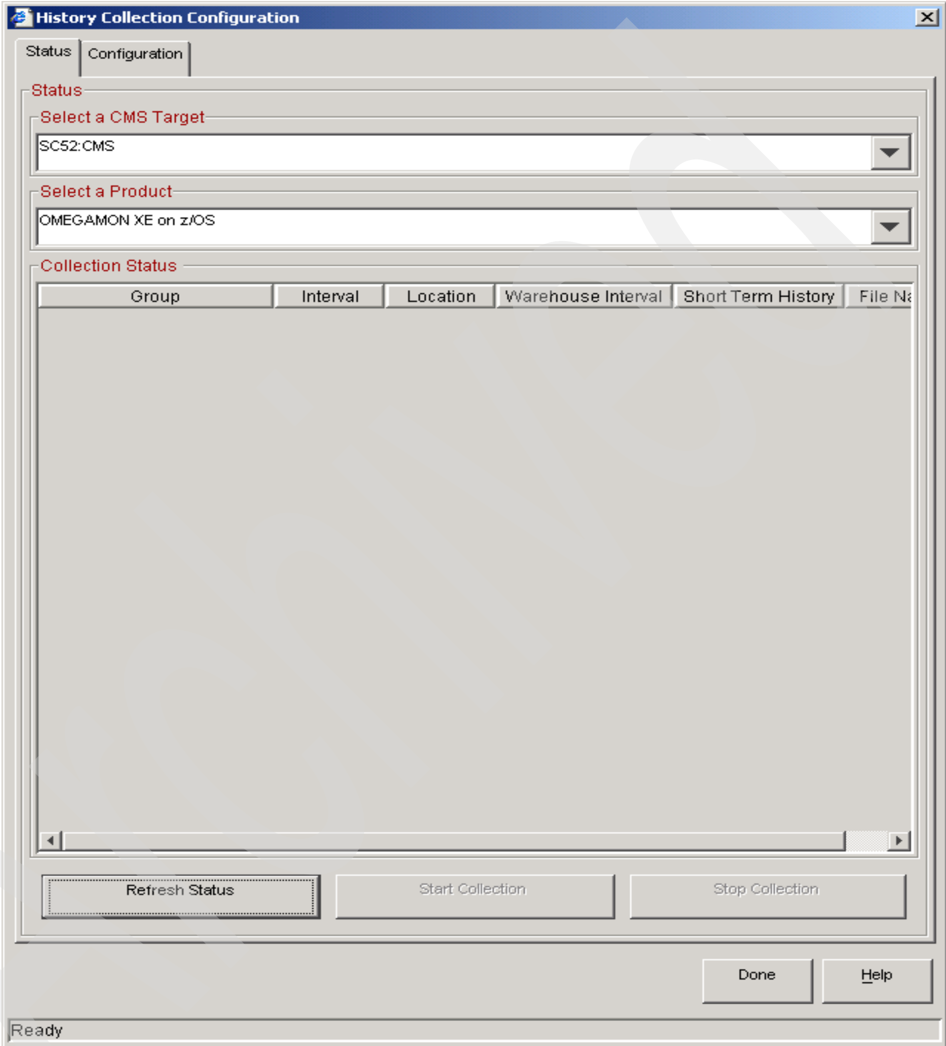


Figure 6-25 Historical collection configuration - initial status

To define a new historical collection, switch to the Configuration tab as shown in Figure 6-26 on page 224.

1. Select a product from the list of product groups that you want to configure for historical data collection. (We selected **Address_space_real_storage**.)
2. In Configuration Controls, specify the rules of historical data collection:

   Collection interval The interval at which data is retrieved.

   Collect at Whether the collection is performed by Tivoli Enterprise Monitoring Agent or Tivoli Enterprise Monitoring Server.

   Warehouse interval The interval at which the historical data is sent to the Tivoli Data Warehouse.

   Short Term History The period at which historical data is converted from binary to flat files (default is one day). If you choose not to move data to Tivoli Data Warehouse, the data will be lost after the period.
3. Click **Configure Group (s)** to activate it.

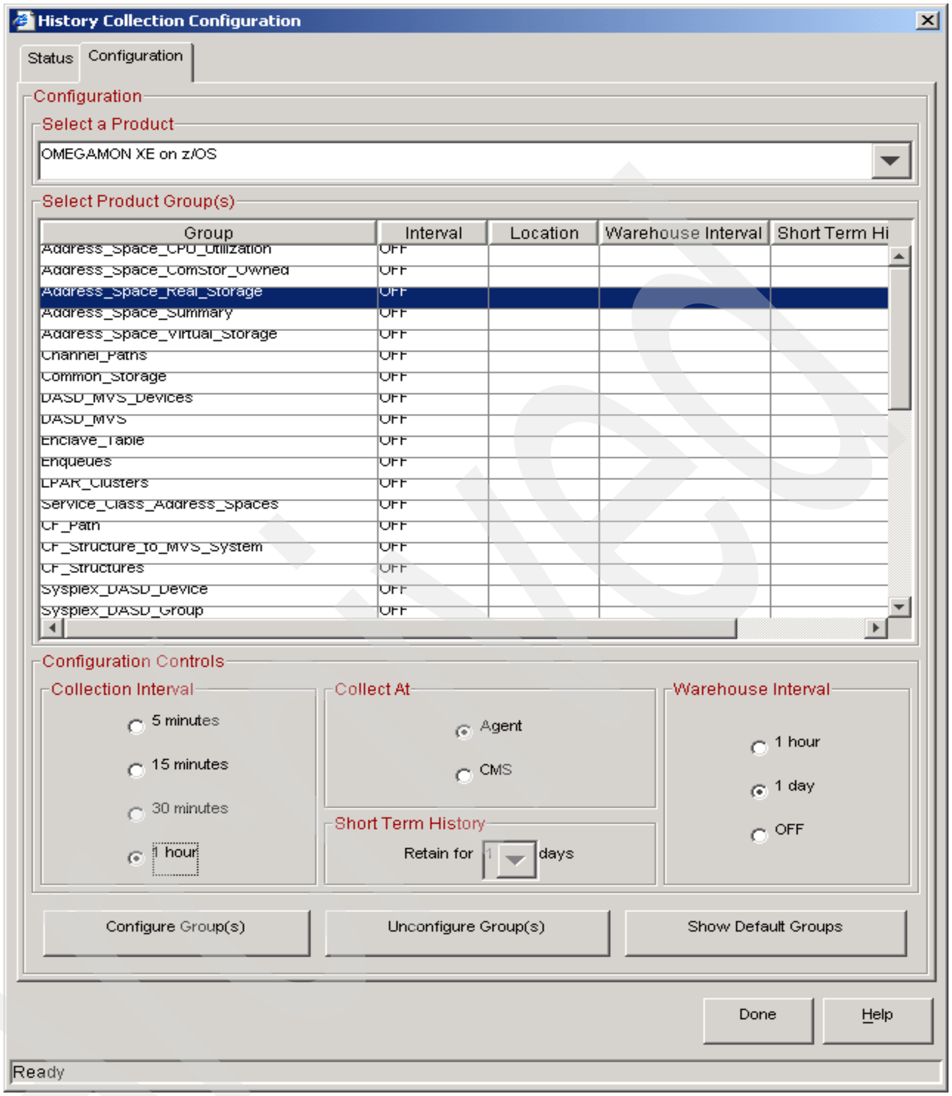


Figure 6-26 Historical collection configuration - defining a collection group

4. Before data is collected, you must start the collection from the Status tab. In the Status tab, click **Start Collection** to start the historical collection for Address_Space_Real_Storage group.
5. Click **Refresh Status**. You can see the collection status for the newly added Address_Space_Real_Storage group as shown in Figure 6-27 on page 225.

The last column of the collection status display File Name shows the name of the file where data will be stored for up to 24 hours.

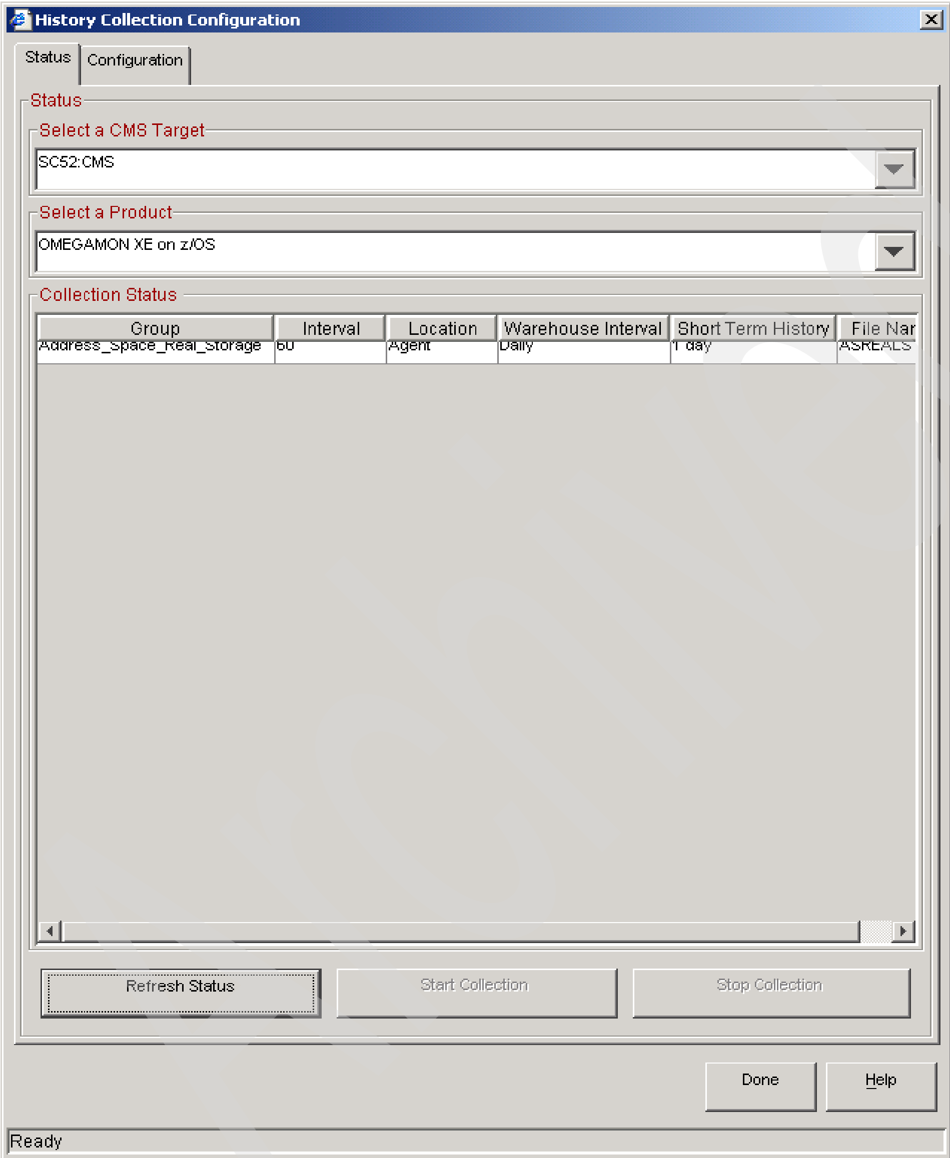


Figure 6-27 Historical data collection - starting collection

6. You can stop the historical data collection for Address_Space_Real_Storage group by clicking **Stop Collection**.

6.6 IBM Tivoli OMEGAMON XE on z/OS new features

This section illustrates some of the new features that are available with IBM Tivoli OMEGAMON XE on z/OS V3.1.0. The discussion includes use of the INSPECT function, followed by 6.6.2, “Reviewing work running on a zAAP processor” on page 229.

6.6.1 Using the INSPECT function with Tivoli Enterprise Portal

The INSPECT function from the OMEGAMON for MVS classic and CUA 3270 implementations has been migrated to the OMEGAMON environment. This function samples the location of the instruction address register in the target address space on a rapid basis for a relatively short period of time. This enables it to build up a profile of which instructions, which programs, and which tasks are consuming CPU resources in the address space. This information can help you identify inefficient code, or where in an address space code may be looping.

In our scenario we investigate the Tivoli Enterprise Monitoring Server address space running on SC67 LPAR.

The INSPECT function is available from the Address Space CPU Utilization workspace. To get to the Address Space CPU Utilization workspace, use the Address Space Overview workspace, which is located in the Navigator tree shown in Figure 6-28.

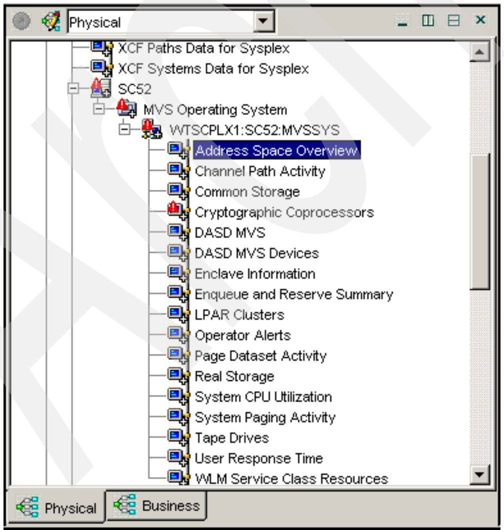


Figure 6-28 Navigator for Address Space Overview workspace

The Address Space CPU Utilization Summary is linked from the Address Space Counts portlet as shown in Figure 6-2 on page 195. Right-click the **Address Space Count** link and select **Address Space CPU Utilization**.

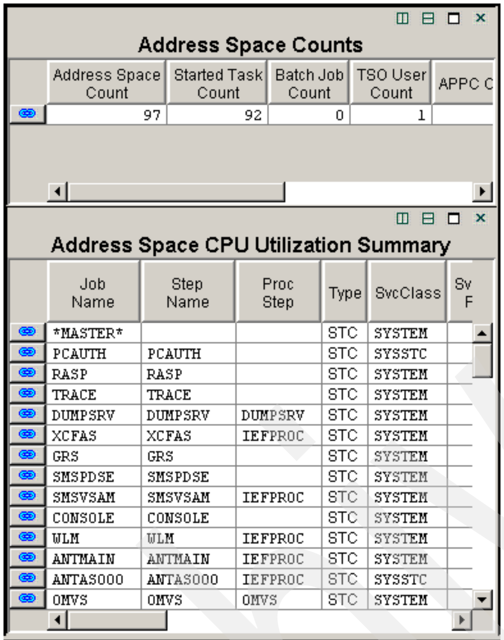


Figure 6-29 Address space overview

The Address Space CPU Utilization workspace shows a list of running address spaces. Right-click an address space to inspect it (in our case CANSDSST), as shown in Figure 6-30. Select **Inspect Address Space Cpu Use**.

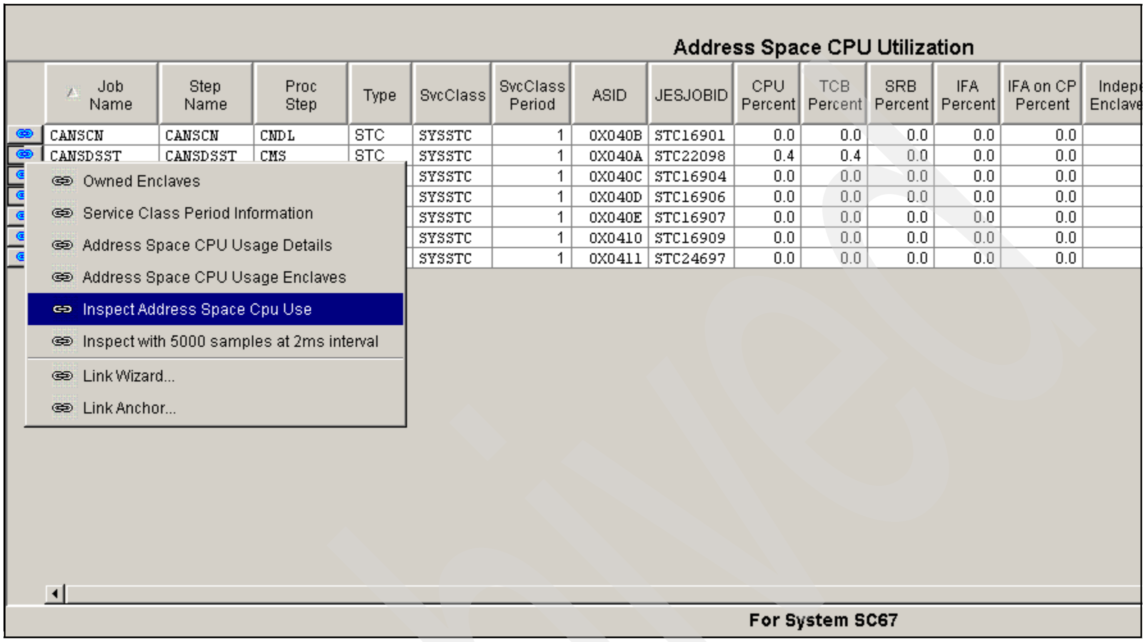
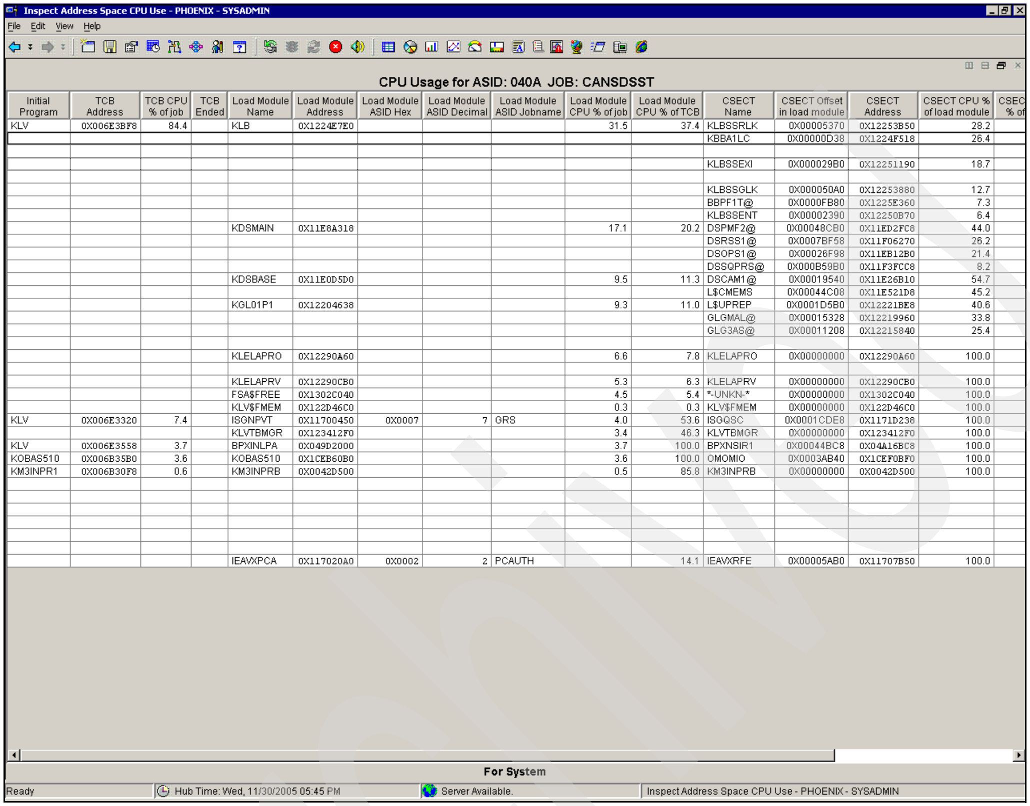


Figure 6-30 Address space link screen

Figure 6-31 shows the INSPECT detail for address space CANSDSST, our Tivoli Enterprise Monitoring Server running on SC67.



Initial Program	TCB Address	TCB CPU % of job	TCB Ended	Load Module Name	Load Module Address	Load Module ASID Hex	Load Module ASID Decimal	Load Module ASID Jobname	Load Module CPU % of job	Load Module CPU % of TCB	CSECT Name	CSECT Offset in load module	CSECT Address	CSECT CPU % of load module	CSEC % of
KLV	0X006E3BF8	84.4		KLB	0X1224E7E0				31.5	37.4	KLBSSRLK	0X00005370	0X12253B50	28.2	
											KBBA1LC	0X00000D38	0X1224F518	26.4	
											KLBSSEXI	0X000029B0	0X12251190	18.7	
											KLBSSGLK	0X000050A0	0X12253880	12.7	
											BBPF1T@	0X0000FB80	0X1225E360	7.3	
											KLBSSENT	0X00002390	0X12250B70	6.4	
				KDSMAIN	0X11E8A318				17.1	20.2	DSPMF2@	0X00048CB0	0X11ED2FC8	44.0	
											DSRSS1@	0X0007BF58	0X11F06270	26.2	
											DSOPS1@	0X00026F98	0X11EB12B0	21.4	
											DSSQPRS@	0X000B59B0	0X11F3FCC8	8.2	
				KDSBASE	0X11E0D5D0				9.5	11.3	DSCAM1@	0X00019540	0X11E26B10	54.7	
											L$CMEMS	0X00044C08	0X11E521D8	45.2	
				KGL01P1	0X12204638				9.3	11.0	L$UPREP	0X0001D5B0	0X12221BE8	40.6	
											GLGMAL@	0X00015328	0X12219960	33.8	
											GLG3AS@	0X00011208	0X12215840	25.4	
				KLELAPRO	0X12290A60				6.6	7.8	KLELAPRO	0X00000000	0X12290A60	100.0	
				KLELAPRV	0X12290CB0				5.3	6.3	KLELAPRV	0X00000000	0X12290CB0	100.0	
				FSA$FREE	0X1302C040				4.5	5.4	*-UNKN-*	0X00000000	0X1302C040	100.0	
				KLV$FMEM	0X122D46C0				0.3	0.3	KLV$FMEM	0X00000000	0X122D46C0	100.0	
KLV	0X006E3320	7.4		ISGNPVT	0X11700450	0X0007	7	GRS	4.0	53.6	ISGQSC	0X0001CDE8	0X1171D238	100.0	
				KLVTBMGR	0X123412F0				3.4	46.3	KLVTBMGR	0X00000000	0X123412F0	100.0	
KLV	0X006E3558	3.7		BPXINLPA	0X049D2000				3.7	100.0	BPXNSIR1	0X00044BC8	0X04A16BC8	100.0	
KOBAS510	0X006B35B0	3.6		KOBAS510	0X1CEB60B0				3.6	100.0	OMOMIO	0X0003AB40	0X1CEF0BF0	100.0	
KM3INPR1	0X006B30F8	0.6		KM3INPRB	0X0042D500				0.5	85.8	KM3INPRB	0X00000000	0X0042D500	100.0	
				IEAVXPCA	0X117020A0	0X0002	2	PCAUTH		14.1	IEAVXRFE	0X00005AB0	0X11707B50	100.0	

Figure 6-31 INSPECT data for CANSDSST

6.6.2 Reviewing work running on a zAAP processor

IBM Tivoli OMEGAMON XE on z/OS has added support for zSeries Application Assist Processors (zAAPs). zAAPs are a special class of assist processor designed to run Java workloads.

For reporting purposes, a zAAP is usually referred to as an Integrated Facility for Applications (IFA). You can force Java workloads to execute only on the IFAs, maximizing the offload of work from regular engines, or allow them to execute on central processors (CPs) when they are not busy. This is known as crossover.

If you allow workloads to run on regular central processors, you can allow them to run at the priority of the service class (dispatch priority honored) or at a lower priority (dispatch priority not honored).

For IBM Tivoli OMEGAMON XE for on z/OS, find details by starting the Address Space CPU Overview workspace (Figure 6-29 on page 227) and following the link to the Address Space CPU Utilization. You can also drill down into a specific address space and get the Address Space CPU Usage Details workspace. We use the BDAGNTA started task, which is the WebSphere agent that runs Java processing. Figure 6-32 shows the resulting Address Space CPU Usage Details workspace. Note the IFA PERCENT and IFA CPU PERCENT columns.

Address Space CPU Usage Details

Job Name	Step Name	ASID	Type	CPU Percent	TCB Percent	SRB Percent	IFA Percent	IFA on CP Percent	Job CPU Percent	Job TCB Percent	Job SRB Percent	Job Additional SRB Service Percent	Job Preemptable Home SRB Service Percent	Job CPU Time
BDAGNTA	BDAGNTA	0X0065	STC	0.4	0.4	0.0	0.1	0.1	0.4	0.3	0.1	0.0	0.0	4,478.73

For System SC52

Figure 6-32 IFA (zAAP) usage

6.7 Sysplex-related monitoring

This section discusses some monitoring related to IBM Tivoli OMEGAMON XE on z/OS in a sysplex environment:

- 6.7.1, “Enqueue Plex” on page 230
- 6.7.2, “Sysplex proxy” on page 233
- 6.7.3, “DASD threshold filtering” on page 236

6.7.1 Enqueue Plex

An environment with multiple sysplexes may or may not share DASD. In fact, shared DASD predates the concept of a sysplex, thus many sites have DASD shared between development and production sysplexes or between separate production sysplexes. This requires a mechanism to protect against two applications executing in different sysplexes from attempting to write to the same file at the same time. It is true that hardware path contention would prevent simultaneous writes, but it would not prevent interleaved writes, which might compromise the integrity of the data.

IBM Tivoli OMEGAMON XE for z/OS uses the IBM GQSCAN facility to gather enqueue information. This facility does not directly indicate when cross-sysplex conflicts occur. As such, IBM Tivoli OMEGAMON XE for z/OS provides an *ENQplex* facility to handle this scenario.

IBM Tivoli OMEGAMON XE for z/OS uses the term ENQplex as an identifier to denote the boundaries of DASD sharing. Therefore, two or more sysplexes that share DASD would be assigned to the same ENQplex, while a single sysplex with its own DASD would be assigned to it own unique ENQplex. The assignment of sysplexes to specific ENQplexes clarifies the issue of resource ownership and contention. For example, naming explicit DASD sharing boundaries enables you to identify actual conflicts for a resource.

IBM Tivoli OMEGAMON XE for z/OS collects the conflict information from each sysplex in the ENQplex and correlate the owner information to provide the Global Enqueue and Reserve information.

The example in Figure 6-33 indicates that ENQplex ITSOENQ currently contains one sysplex and three runtime environments (RTEs) representing the three images in sysplex.

```
---------------------- Define Enterprise wide ENQ manager ---------------------
COMMAND ===>

 Action: A Add a new ENQPlex environment
         D Delete an ENQPlex environment
         V View list of Sysplexs assigned to an ENQPlex

Action ENQPlex                                          Sysplex    RTE
code   name      Description                             count    count

 _     ________  ______________________________________
 _     $DEFAULT  DEFAULT ENQPLEX ENVIRONMENT               0        0
 _     ITSOENQ   ITSO Enqueue Plex                         1        3
 F1=Help  F3=Back  F7=Up  F8=Down
```

Figure 6-33 Defining ENQ scope

By viewing using the `V` action about the ENQplex ITSOENQ, you see the name of the sysplexes within this ENQplex. At this time, there is only one sysplex in this ENQplex: WTSCPLX1, as shown in Figure 6-34 on page 232.

```
---------------- View list of Sysplexs assigned to an ENQPlex ----------------
COMMAND ===>

Current ENQPlex:   ITSOENQ
Description:       ITSO Enqueue Plex

Sysplex   Sysplex                                  RTE
name      description                             count
WTSCPLX1  ITSO Plex                                 3
 F1=Help  F3=Back  F7=Up  F8=Down
```

Figure 6-34 Viewing members of ENQ plex

Figure 6-35 shows the Global Enqueue Data for Sysplex information provided by IBM Tivoli OMEGAMON XE for z/OS. In the example, you see that there is currently one enqueue conflict across the ENQplex and sysplex. The Global Enqueue Data view of the workspace Global Enqueue Data for Sysplex shows two tasks waiting for an enqueue on a resource with major name KLVGLOCK and minor name 1WTSCPLX. By selecting the Link symbol you can drill down to additional information about the enqueues.

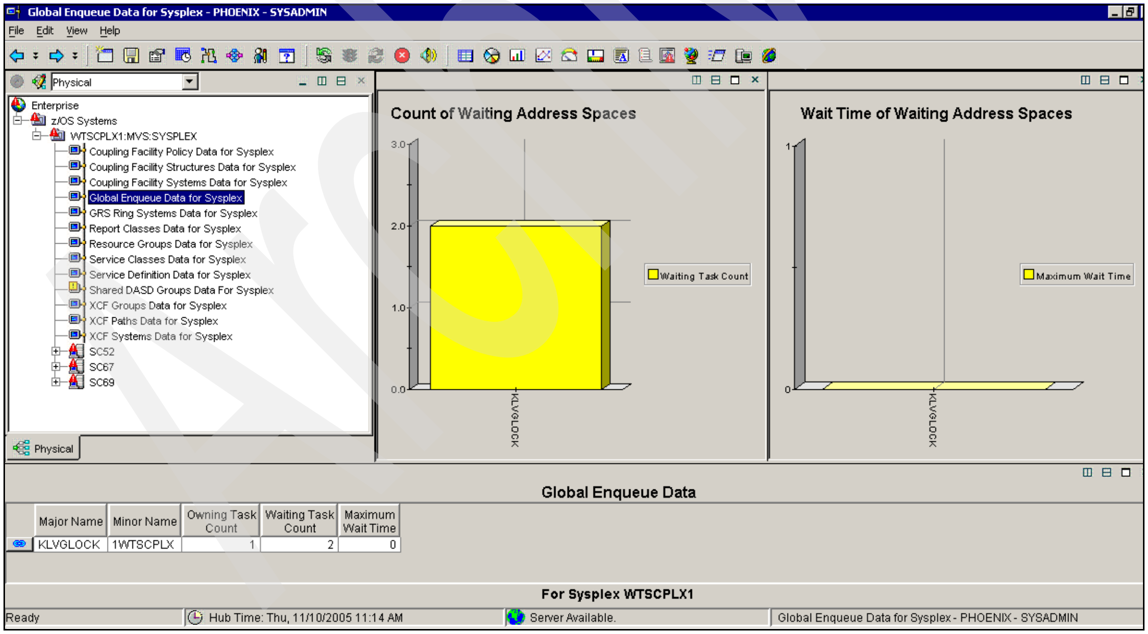


Figure 6-35 Global Enqueue Data for Sysplex

Figure 6-36 shows the Global Enqueues and Reserves workspace with additional details about the enqueue conflict. Address space CANSDS52 on system SC52 has an exclusive enqueue on a resource. Address spaces on systems SC67 and SC69, CANSDSST in both cases, are requesting exclusive enqueues as well.

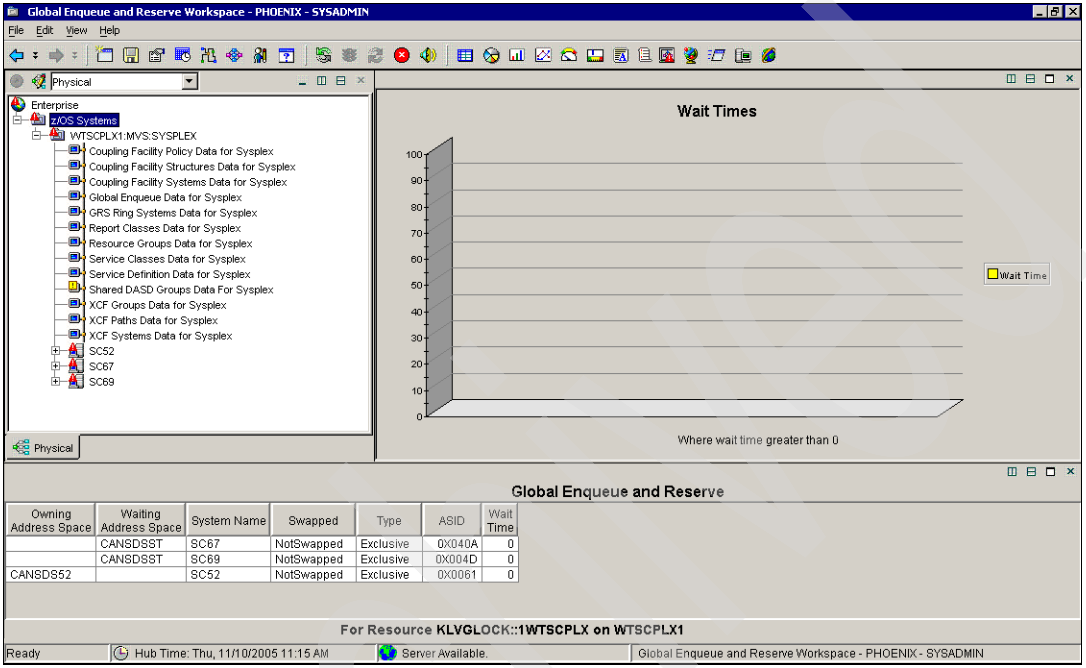


Figure 6-36 Global enqueues and reserves

> **Note:** The enqueue detailed in the previous figures is part of the sysplex proxy mechanism described in a subsequent section. This enqueuing scheme is used to control which Tivoli Enterprise Monitoring Server is acting as the sysplex proxy for a given sysplex.

6.7.2 Sysplex proxy

IBM Tivoli OMEGAMON XE for z/OS provides detailed information about the status of your sysplex. IBM Tivoli OMEGAMON XE for z/OS enables you to monitor one or more sysplexes. Within the scope of an individual sysplex, the objects monitored at the component level include workloads, Coupling Facility, cross-system Coupling Facilities, global enqueues, shared DASD, and GRS rings.

Virtually all the sysplex metrics provided by IBM Tivoli OMEGAMON XE for z/OS are derived by combining observations made on each z/OS image in the sysplex.

- Shared DASD metrics combine I/O data from all images to determine delay due to other images and total I/O rate for a device
- Workload Manager service class period transaction counts are combined to get sysplex-wide response times and average time per transaction and to determine the sysplex-wide performance index for each service class period
- Coupling Facility I/O activity is accumulated for each structure

When configuring IBM Tivoli OMEGAMON XE for z/OS, one Tivoli Enterprise Monitoring Server in every sysplex must be selected as the primary focal point for data originating from the sysplex. This Tivoli Enterprise Monitoring Server performs the sysplex proxy function. The sysplex proxy Tivoli Enterprise Monitoring Server calculates performance numbers for the sysplex, collects and stores data for the sysplex, and evaluates sysplex-wide situations. The sysplex proxy can reside in either a remote or hub Tivoli Enterprise Monitoring Server. The sysplex proxy should be hosted in a remote Tivoli Enterprise Monitoring Server. You typically exclude the hub Tivoli Enterprise Monitoring Server from becoming a sysplex proxy. Because of the importance and necessity of this function, every Tivoli Enterprise Monitoring Server in the sysplex except those specifically excluded are designated as a backup for the primary sysplex proxy.

Figure 6-37 shows the definition of our sysplex, WTSCPLX1. We select the `V` action code to view the RTEs.

```
----------------------- Define your Sysplex environment -----------------------
COMMAND ===>

Action: A Add a new Sysplex
          Important: This name does not have to be the actual Sysplex name.
                     Do not repeat a name already in use by another
                     implementation of this product on the same IBM Sysplex.
        U Update a Sysplex
        D Delete a Sysplex
        V View list of RTEs assigned to a Sysplex

Action Sysplex   ENQPlex                                            RTE   Use
code   name      name      Description                             count  name
 _     ________
 _     WTSCPLX1  ITSOENQ   ITSO Plex                                 3     Y
 F1=Help  F3=Back  F7=Up  F8=Down
```

Figure 6-37   Define sysplex environment

Figure 6-38 provides details about the three RTEs for our z/OS images.

```
---------------------- List of RTEs assigned to a Sysplex ---------------------
COMMAND ===>

 Current Sysplex:     *WTSCPLX1
 * Use WTSCPLX1 Sysplex name when building managed system name ==> Y (Y, N)
 Sysplex Desc:         ITSO Plex
 Current ENQPlex:      ITSOENQ
 ENQPlex Desc:         ITSO Enqueue Plex
 Datastore high-level qualifier: OMEGAMON.WTSCPLX1

RTE       RTE                                   Proxy      CMS
name      description                           position   type
SHR1      Sharing RTE for SC52                  PRIMARY    HUB
SHR2      Sharing RTE for SC69                  BACKUP     REMOTE
SHR3      Sharing RTE for SC67                  BACKUP     REMOTE
 F1=Help  F3=Back  F7=Up  F8=Down
```

Figure 6-38 RTEs assigned to sysplex

The following general rules should be considered when determining the sysplex proxy eligibility and exclusions:

- Exclude the hub Tivoli Enterprise Monitoring Server from sysplex proxy eligibility. During installation and configuration activities you may have configured the hub Tivoli Enterprise Monitoring Server first, so the hub likely defaulted to be the primary sysplex proxy. After defining the first remote Tivoli Enterprise Monitoring Server, you should modify the hub Tivoli Enterprise Monitoring Server RTE and exclude it from sysplex proxy eligibility.
- If you have capacity-constrained LPARs, consider excluding Tivoli Enterprise Monitoring Servers on these images from sysplex proxy eligibility.
- When taking a sysplex-wide IBM Tivoli OMEGAMON XE for z/OS outage, shut down Tivoli Enterprise Monitoring Server started tasks in an order such that you do not force a sysplex proxy failover. Stop the current sysplex proxy Tivoli Enterprise Monitoring Server last.
- The only way to cause a sysplex proxy failover is to shut down the Tivoli Enterprise Monitoring Server currently acting as the sysplex proxy. Additionally, there is no mechanism to force the sysplex proxy back to the original Tivoli Enterprise Monitoring Server other than terminating all of the other eligible Tivoli Enterprise Monitoring Server address spaces.

The first Tivoli Enterprise Monitoring Server address space that is sysplex proxy eligible obtains an enqueue on resource KLVGLOCK:{Plex name} as seen in

Figure 6-35 on page 232. The RKLVLOG for this Tivoli Enterprise Monitoring Server instance will contain the KOS101I message:

```
KOSKFA00: KOS101I  CMS SC52:CMS IS NOW THE SYSPLEX PROXY
```

As other Tivoli Enterprise Monitoring Server address spaces are started, they will wait on the KLVGLOCK:{Plex name} resource as shown in Figure 6-36 on page 233. Therefore, if the current sysplex proxy Tivoli Enterprise Monitoring Server is stopped, the first waiting Tivoli Enterprise Monitoring Server will acquire the exclusive enqueue and assume the sysplex proxy function.

Sysplex proxy recovery

If the sysplex proxy Tivoli Enterprise Monitoring Server fails, the "next" Tivoli Enterprise Monitoring Server in the sysplex takes on the sysplex proxy status. "Next" is determined by the start-up order of the CMSs within the scope of the CMSs eligible (that is, not excluded) to perform sysplex proxy functions. The other CMSs will redirect their information to the new sysplex proxy. Restart of the original sysplex proxy does not modify the sysplex proxy processing.

Historical collection

The sysplex proxy Tivoli Enterprise Monitoring Server collects historical data to Persistent Data Stores (PDS) files. If you have EPILOG historical collector configured, it will also collect sysplex history. If the sysplex proxy fails over into another Tivoli Enterprise Monitoring Server, the PDS files will be opened by the new Tivoli Enterprise Monitoring Server to resume historical collection. However, the EPILOG collector on the new Tivoli Enterprise Monitoring Server will not be aware of these change. EPILOG must be recycled to activate historical collection for sysplex information.

6.7.3 DASD threshold filtering

When a sysplex has a large number of images, and a large number of DASD devices are shared, the volume of data that is normally sent to the sysplex proxy can be extensive. The Candle Technology (CT) facility that processes the data could be overwhelmed. The result is high CPU consumption and an apparent storage creep condition.

In reality, most of the DASD metrics being collected are unremarkable. That is, the information represents normal, well-behaved DASD activity rather than abnormal conditions. DASD Threshold Filtering provides a mechanism for indicating which devices are important, then collects metrics for just these devices from all images. DASD Threshold Filtering uses the Situation Event process to specify devices of interest.

Using the situation editor, you specify the inclusion criteria and the interval. The inclusion criteria leads to the construction of a volume list that is sent to each Tivoli Enterprise Monitoring Server, which should reduce the amount of volume-related data. As a general rule, the DASD Threshold Filtering situation should be simple and reasonable in order to achieve the best performance.

- Avoid using the attributes System or Group Name.
- The sampling interval is a balance between CPU consumption and data fidelity. A shorter sampling interval will catch out-of-norm devices more quickly at the cost of higher CPU consumption, but a longer sampling interval may miss some devices of interest but require fewer CPU resources. Start by using a sampling interval around 30 minutes and adjust it as required.
- If a new Tivoli Enterprise Monitoring Server is added to a sysplex or a Tivoli Enterprise Monitoring Server is recycled, stop and restart the DASD Filtering Threshold situation. The device list constructed by the situation is distributed to each z/OS image and maintained in storage. If the z/OS image comes online after the list has been distributed, this image will send all of its DASD information until the next volume list is built. Recycling the situation will reset the list on all images.

Figure 6-39 shows a simple DASD Filtering Threshold situation named DASD_Filtering that limits the data collection to devices that have I/O activity (I/O Rate greater than 0.0).

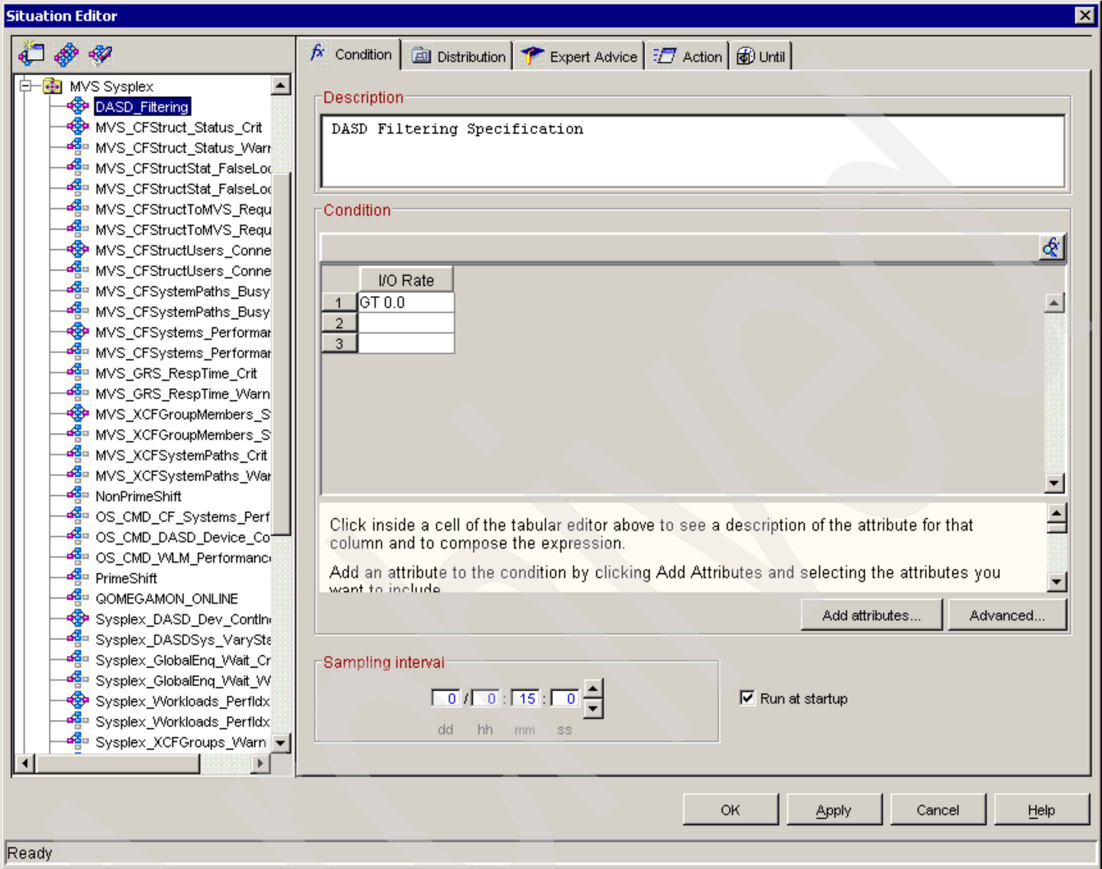


Figure 6-39 DASD Threshold Filtering situation

Our DASD filtering situation uses the DASD_Device_Collection_Filtering attribute group. Figure 6-40 shows this group's attributes. Most available DASD-related attributes are eligible to be used in the specification.

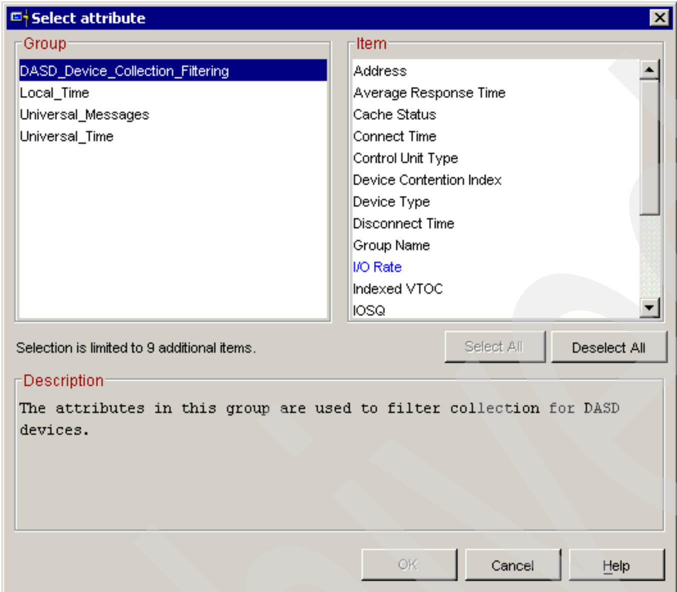


Figure 6-40 DASD filtering attributes

DASD Threshold Filtering runtime messages

When Tivoli Enterprise Monitoring Server or the DASD Threshold Filtering situation is started, a KO46256 message appears in the RKLVLOG of the Tivoli Enterprise Monitoring Server:

```
KO46256    Situation definition DASD_Filtering created by *ENTERPRISE.
```

Our WTSCPLX1 sysplex has approximately 5700 volumes. The messages in Example 6-1 show the DASDThreshold Filtering criteria specified above reduced the amount of data from metrics from 5700 volumes to metrics for less than 100 volumes on each sampling interval.

Example 6-1 DASD Threshold Filtering volume counts

```
KOSDCMML Threshold filtering true for 55 DASD device volumes.
KOSDCMML Threshold filtering true for 93 DASD device volumes.
KOSDCMML Threshold filtering true for 23 DASD device volumes.
KOSDCMML Threshold filtering true for 93 DASD device volumes.
KOSDCMML Threshold filtering true for 23 DASD device volumes.
KOSDCMML Threshold filtering true for 93 DASD device volumes.
KOSDCMML Threshold filtering true for 24 DASD device volumes.
```

6.8 CICS Short on Storage (SOS)

You can use IBM Tivoli OMEGAMON XE for CICS on z/OS in quite a few common CICS situations. Our first scenario shows the screen reporting an SOS (short on storage) condition with EDSA. In our CICS log we see this message:

```
DFHSM0133 SCSCBUD1 CICS is under stress (short on storage above 16MB).
```

Our OMEGAMON for CICS shows the condition in the storage workspace or using a predefined situation. Figure 6-41 depicts the default DSA display showing the condition.

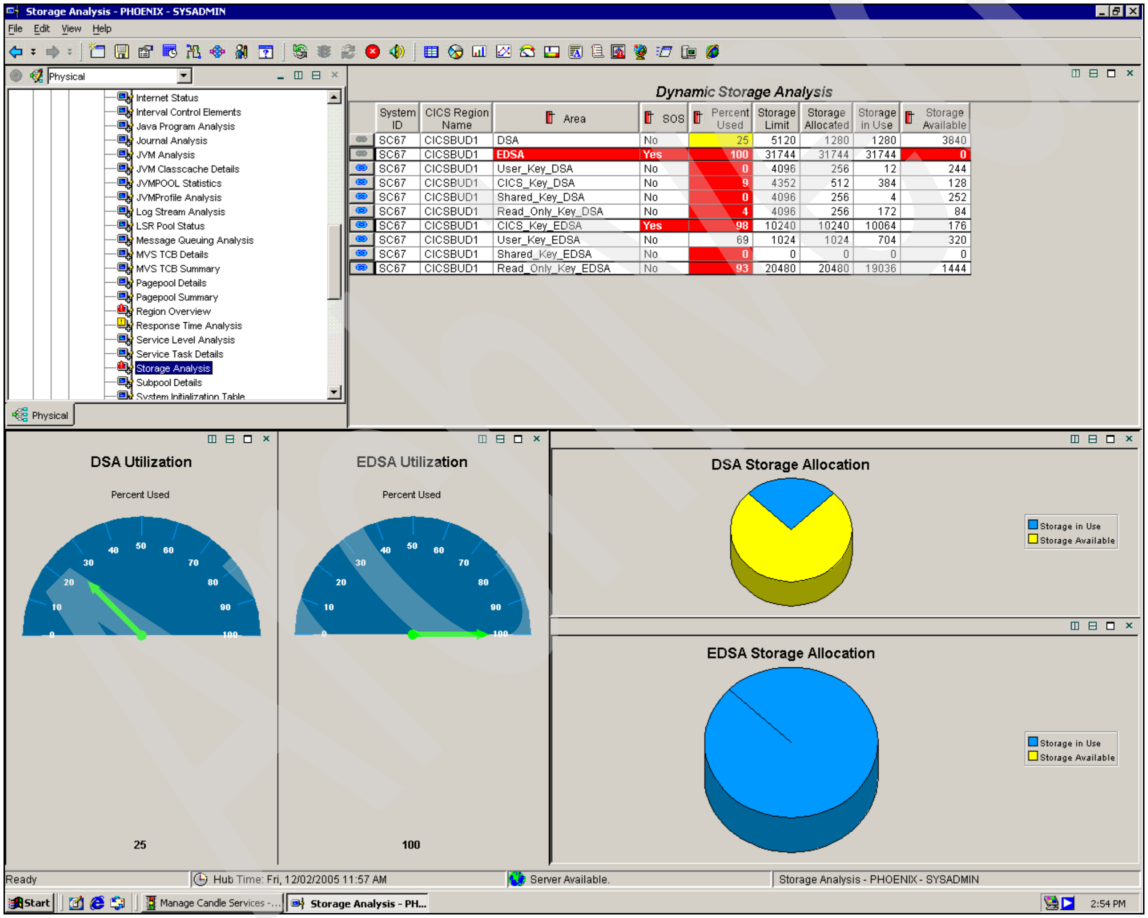


Figure 6-41 CICS Dynamic Storage Display during SOS period

A

Product codes

This appendix contains a list of the product codes used by the IBM Tivoli products in messages and modules. It is not an exhaustive list, and is subject to change when products are re-versioned or are no longer supported. The table is provided to help you diagnose components involved.

Product codes

The list in Table A-1 may not be all-inclusive of the internal component codes for former Candle products. The code is prefixed with the letter K.

Table A-1 Candle product codes

Product name	CODE
OMEGAMON XE/DE for OS/400®	A4
AF/OPERATOR®	AO
AF/REMOTE® on Windows NT®	AT
Integrated Resource Manager	AU
OMEGAMON II for CICS	C2
OMEGAMON XE/DE for CICS on z/OS	C5
Configuration tool (z/OS Installer)	CI
CandleNet Portal Server	CJ
OMEGAMON DE	CJ
OMEGAMON XE/DE for CICS on z/OS	CP
OMEGAMON II for DB2	D2
OMEGAMON XE/DE for DB2 on z/OS	D5
OMEGAMON II for Storage	DF
OMEGAMON XE/DE for DB2plex	DP
Candle Management Server	DS
Candle Dump Analysis Tool for z/OS	DZ
SNMP Gateway on Windows NT	EN
OMEGAMON II for IMS	I2
OMEGAMON II for DBCTL	ID
IOMAN FOR DB2	IO
OMEGAMON for IMS on z/OS	I5
CL/CONFERENCE®	LC
CL/SUPERSESSION®	LS

Product name	CODE
OMEGAMON II CUA and TEMS Component	LV
OMEGAMON XE/DE for Linux	LZ
OMEGAMON II for MVS	M2
OMEGAMON XE/DE for z/OS	M5
CCC for MQSeries Configuration	MC
PQEDIT/MVS	MD
OMEGAMON XE/DE for WebSphere MQ	MQ
OMEGAVIEW	MV
Candle Management Workstation	MW
OMEGAMON XE/DE for Mainframe Networks	N3
OMEGAMON XE/DE for Windows NT/2000	NT
Alert Adapter for TME® 10™ NetView	NV
OMEGAMON XE/DE for Netware	NW
OMEGAMON II for DB2 Classic	O2
Omnimon Base (Component of OMEGAMON Classics)	OB
OMEGAMON II for CICS Classic	OC
OMEGAMON XE/DE for UNIX System Services (USS)	OE
OMEGACENTER GATEWAY for MVS	OG
OMEGAMON II for IMS Classic	OI
OMEGAMON II for MVS Classic	OM
OMEGAMON II for Mainframe Networks	ON
OMEGAMON XE/DE for MS SQL Server	OQ
OMEGAMON XE/DE for Oracle	OR
OMEGAMON XE/DE for Sysplex	OS
OMEGAMON for VM	OV
OMEGAMON XE/DE for Sybase	OY
OMEGAMON XE/DE for WebSphere MQ Integrator	QI

Product name	CODE
AF/REMOTE on Windows NT	RP
OMEGAMON XE/DE for Storage	S3
OMEGAMON XE/DE for R/3	SA
DB/QUICKCOMPARE	TA
DB/DASD	TD
DB/EXPLAIN	TE
OMEGAMON Alert Manager for CA-Unicenter TNG	TN
DB/QUICKCHANGE	TQ
DB/SMU	TS
OMEGAMON Alert Manager for Tivoli/Enterprise Console	TV
DB/WORKBENCH (OK to ship standalone)	TW
OMEGAMON XE/DE for Tuxedo	TX
OMEGAMON XE/DE for DB2 Universal Database™	UD
Universal Agent	UM
OMEGAMON XE/DE for UNIX	UX
IntrlliWatch Analyzer	W2
OMEGAMON for WebSphere App. Server (Distributed)	WE
IntelliWatch® Monitor	WM
OMEGAVIEW II® for the Enterprise	WO
OMEGAMON DE for OS/390	WO
OMEGAMON for WAS for OS/390	WW
CandleNet eBusiness Platform®	XC

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 249. Note that some of the documents referenced here may be available in softcopy only.

- *Getting Started with IBM Tivoli Monitoring Version 6.1*, SG24-7143
- *IBM Tivoli Composite Application Manager V6.0 Family: Installation, Configuration, and Basic Usage*, SG24-7151
- *Implementing IBM Tivoli OMEGAMON XE for WebSphere Business Integration V1.1*, SG24-6768

Other publications

These publications are also relevant as further information sources:

- IBM Tivoli OMEGAMON XE documentation
  - *Using OMEGAMON(R) Products: CandleNet Portal*, GC32-9182
  - *Administering OMEGAMON Products: CandleNet Portal*, GC32-9180
  - *Historical Data Collection Guide for IBM Tivoli OMEGAMON XE Products*, GC32-9429
  - *Installing and Setting Up OMEGAMON Platform and Candlenet Portal on Windows and UNIX*, SC32-1768
  - *Configuring IBM Tivoli Candle Management Server on z/OS, Version 360*, GC32-9414
  - *IBM Tivoli Candle Product Messages, Volume 1*, SC32-9416
  - *IBM Tivoli Candle Product Messages, Volume 2*, SC32-9417
  - *IBM Tivoli Candle Product Messages, Volume 3*, SC32-9418
  - *IBM Tivoli Candle Product Messages, Volume 4*, SC32-9419
  - *IBM Tivoli Candle Product Messages, Volume 5*, SC32-9420

- IBM Tivoli OMEGAMON XE on z/OS V3.1.0 manuals:
  - *Getting Started with IBM Tivoli OMEGAMON XE on z/OS V3.1.0*, SC32-9491
  - *Configuring IBM Tivoli OMEGAMON XE on z/OS V3.1.0*, SC32-9364
  - *Using IBM Tivoli OMEGAMON XE on z/OS V3.1.0*, GC32-9209
  - *IBM Tivoli OMEGAMON XE on z/OS Release Notes V3.1.0*, GI11-4038
  - *OMEGAMON II for MVS Configuration and Customization Guide V5.5.0*, GC32-9277
  - *OMEGAMON II for MVS User's Guide, V5.5.0*, GC32-9280
  - *OMEGAMON II for MVS Command Language Reference Manual, V5.5.0*, GC32-9276
  - *EPILOG for MVS Command Language Reference Manual, V5.5.0*, GC32-9265
  - *IBM Tivoli End-to-End Response Time Feature (ETE) Reference Manual, v5.5.0*, SC32-9376
  - *IBM Tivoli OMEGAMON XE on z/OS Program Directory*, GI11-4081
- IBM Tivoli OMEGAMON XE for Storage on z/OS V3.1.0 manuals:
  - *OMEGAMON II for SMS Administrator's Guide*, GC32-9281
  - *Configuring IBM Tivoli OMEGAMON XE for Storage on z/OS and OMEGAMON II for SMS*, SC32-9371
  - *IBM Tivoli OMEGAMON XE for Storage on z/OS: Getting Started*, SC32-9411
  - *Tuning IBM Tivoli OMEGAMON XE for Storage on z/OS and OMEGAMON II for SMS*, SC32-9396
  - *Using IBM Tivoli OMEGAMON XE for Storage on z/OS and OMEGAMON II for SMS*, SC32-9470
  - *IBM Tivoli OMEGAMON XE for Storage on z/OS: Release Notes*, GI11-4036
  - *IBM Tivoli OMEGAMON XE for Storage Program Directory*, GI11-4079
- IBM Tivoli OMEGAMON XE for IMS on z/OS V3.1.0 manuals:
  - *Getting Started with IBM Tivoli OMEGAMON XE for IMS on z/OS*, SC32-9469
  - *Configuring IBM Tivoli OMEGAMON XE for IMS on z/OS*, SC32-9354
  - *Using IBM Tivoli OMEGAMON XE for IMS on z/OS*, GC32-9351
  - *IBM Tivoli OMEGAMON XE for IMS on z/OS Release Notes*, GI11-4037

– *IBM Tivoli OMEGAMON II for IMS User's Guide*, GC32-9355
– *IBM Tivoli OMEGAMON II for IMS Configuration and Customization Guide*, SC32-9356
– *IBM Tivoli OMEGAMON II for IMS Console Facility*, SC32-9357
– *IBM Tivoli OMEGAMON II for IMS Transaction Reporting Facility*, SC32-9358
– *IBM Tivoli OMEGAMON II for IMS Bottleneck Analysis Reference Manual*, SC32-9359
– *IBM Tivoli OMEGAMON II for IMS Historical Component (EPILOG) Reference Manual*, SC32-9360
– *IBM Tivoli OMEGAMON II for IMS Historical Component (EPILOG) User's Guide*, GC32-9361
– *IBM Tivoli OMEGAMON II for IMS Realtime Commands Reference Manual*, SC32-9362
– *IBM Tivoli OMEGAMON II for IMS Response Time Analysis (RTA) Reference Manual*, SC32-9363
– *IBM Tivoli OMEGAMON II for IMS Application Trace Facility*, SC32-9470
– *End-to-End Response Time Feature (ETE) Reference Manual*, SC32-9376
– *IBM Tivoli OMEGAMON XE for IMS on z/OS Program Directory*, GI11-4077

- IBM Tivoli OMEGAMON XE for Mainframe Networks V3.1.0 manuals
  – *IBM Tivoli OMEGAMON XE for Mainframe Networks: Getting Started*, GC32-9402
  – *IBM Tivoli OMEGAMON XE for Mainframe Networks: Messages*, GC32-9481
  – *Using IBM Tivoli OMEGAMON XE for Mainframe Networks*, SC32-9405
  – *Configuring IBM Tivoli OMEGAMON XE for Mainframe Networks*, SC32-9403
  – *Configuring OMEGAMON II for Mainframe Networks*, SC32-9404
  – *IBM Tivoli OMEGAMON XE for Mainframe Networks: Release Notes*, GI11-4070
  – *OMEGAMON II for Mainframe Networks User's Guide*, GC32-9274
  – *OMEGAMON II for Mainframe Networks Historical Reporting Guide*, GC32-9273

- *OMEGAMON II for Mainframe Networks NCP Monitoring Guide*, GC32-9272
- *IBM Tivoli End-to-End Response Time Feature Reference Manual*, SC32-9376
- *IBM Tivoli OMEGAMON XE for Mainframe Networks Program Directory*, GI11-4078

- IBM Tivoli Monitoring V6.1 documentation
  - *IBM Tivoli Monitoring Quick Start Guide*, SC32-1802
  - *Introducing IBM Tivoli Monitoring, V6.1.0*, GI11-4071
  - *IBM Tivoli Monitoring User's Guide*, SC32-9409
  - *IBM Tivoli Monitoring Administrator's Guide*, SC32-9408
  - *IBM Tivoli Monitoring Installation and Setup Guide*, GC32-9407
  - *IBM Tivoli Monitoring Problem Determination Guide*, GC32-9458
  - *IBM Tivoli Universal Agent User's Guide*, SC32-9459

Online resources

These Web sites and URLs are also relevant as further information sources:

- IBM Tivoli OMEGAMON XE for CICS on z/OS V3.1 information center

  http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/index.jsp?toc=/com.ibm.omegamon.cics.doc/toc.xml
- IBM Tivoli OMEGAMON XE for IMS on z/OS V3.1 information center

  http://publib.boulder.ibm.com/tividd/td/IBMTivoliOMEGAMONXEforIMS3.1.html
- IBM Tivoli OMEGAMON XE for Mainframe Network V3.1 documentation

  http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/index.jsp?toc=/com.ibm.omegamon.mn.doc/toc.xml
- IBM Tivoli OMEGAMON XE on z/OS V3.1 information center

  http://publib.boulder.ibm.com/tividd/td/IBMTivoliOMEGAMONXEforOS3903.1.html
- IBM Tivoli OMEGAMON XE for Storage on z/OS V3.1 documentation

  http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/index.jsp?toc=/com.ibm.omegamon.stor.doc/toc.xml
- IBM Tivoli Monitoring documentation

  http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/index.jsp?toc=/com.ibm.itm.doc/toc.xml

- Candle installation documentation

  http://publib.boulder.ibm.com/tividd/td/CanInstallDoc/SC32-1768-01/en_US/PDF/SC32-1768-01.pdf

- IBM z/OS Preventive Service Planning technical help database

  https://techsupport.services.ibm.com/server/390.psp390

- Tivoli zSeries SMP/E Zone Creation Samples

  http://www-1.ibm.com/support/docview.wss?rs=660&context=SSZJDU&uid=swg21066230&loc=en_US&cs=utf-8&lang=en

- IBM z/OS PTF download information page (ShopzSeries)

  http://www-1.ibm.com/support/docview.wss?uid=swg21225816&rs=2271

- Tivoli support site for IBM Tivoli OMEGAMON XE patches

  ftp://ftp.software.ibm.com/software/tivoli_support/patches/patches_3.6.0
  ftp://ftp.software.ibm.com/software/tivoli_support/patches/patches_1.9.6

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

Abbreviations and acronyms

ACB	Application Control Block
APAR	Authorized Program Analysis Report
API	Application Programming Interface
APPC	Advanced Program to Program Communication
ASCII	American Standard Code for Information Interchange
CBPDO	Custom Build Product Delivery Offering
CCC	Candle Command Center
CD-ROM	Compact Disk Read-Only Memory
CICAT	Candle Installation and Configuration Assistant Tool
CICS	Customer Information Control System
CMS	Candle Management Server
CMW	Candle Management Workstation
CNP	CandleNet Portal
CNPS	CandleNet Portal Server
CPU	Central Processing Unit
CSA	Common Storage Area
CSI	Consolidated Software Inventory
CT	Candle Technology
CUA	Common User Access
DASD	Direct Access Storage Device
DD	Data Definition
DDDEF	DD Definition
DDR	DASD Dump and Restore
DE	Dashboard Edition
DF/HSM	Data Facility Hierarchical Storage Manager
DF/SMS	Data Facility System Managed Storage
DNS	Domain Name Service
DSA	Data Storage Area
DVIPA	Dynamic Virtual IP Addressing
EBCDIC	Enhanced Binary Coded Decimal Instruction Code
EDSA	Extended DSA
EIB	Enterprise Information Base
ESQA	Extended SQA
ETE	End to end
FMID	Function Modification Identifier
FTP	File Transfer Protocol
GRS	Global Resource Serialization
GTF	Generalized Trace Facility
GUI	Graphical User Interface
HPNS	High Performance Networking Service
HPR	High Performance Routing
HTTP	Hyper-text Transfer Protocol
IBM	International Business Machine Corp.
ICSF	Integrated Cryptographic Service Facility
IFA	Integrated Facility for Applications
IMS	Information Management System
IRA	Intelligent Resource Adapter
IRLM	IMS Resource Locking Manager

ISPF	Interactive System Productivity Facility
ISPF/PDF	ISPF/Program Development Facility
ITSO	International Technical Support Organization
JCL	Job Control Language
LPAR	Logical partition
LSR	Local Shared Resource
LU	Logical Unit
MVS	Multiple Virtual Storage
NCP	Network Control Program
NPM	Network Performance Monitor
ODBC	Open database connectivity
OSA	Open System Adapter
PDS	Partitioned Dataset
PDSE	Partitioned Dataset Extended
PMR	Problem Management Record
PSP	Preventive Service Planning
PSW	Program Status Word
PTF	Program Temporary Fix
RAS	Reliability Availability Servicability
RMF	Resource Monitoring Facility
RTE	Runtime Environment
SDSF	System Display and Search Facility
SMF	System Measurement Facility
SMP/E	System Modification Program Extended
SNA	System Network Architecture
SOA	Service Oriented Architecture
SPUFI	SQL Program using File Input
SQA	System Queue Area
SQL	Structured Query Language
SSL	Secure Socket Layer
STC	Started Task Control
TCP/IP	Transmission Control Protocol, Internet Protocol
TEMS	Tivoli Enterprise Monitoring Server
TSO	Time Sharing Option
UDB	Universal Database
UDP	User Datagram Protocol
URL	Universal Resource Locator
USS	UNIX Systems Services
VM	Virtual Machine
VSAM	Virtual Storage Access Method
VSE	Virtual Storage Extended
VTAM	Virtual Telecommunication Access Method
VTOC	Virtual Table of Content
WLM	Work Load Manager
WTO	Write to Operator
XCF	Extended Coupling Facility

Index

IBM Tivoli OMEGAMON XE V3.1 – Deep Dive on z/OS

(0.5” spine)
0.475”<->0.875”
250 <-> 459 pages

IBM Tivoli OMEGAMON XE V3.1 Deep Dive on z/OS

OMEGAMON architecture discussion and considerations

Using OMEGAMON portal for z/OS performance management

Tuning and managing OMEGAMON solutions

This IBM Redbook describes the IBM Tivoli OMEGAMON XE product family, Version 3.1.0, that runs on z/OS. We start by providing the product history over time and the terminology convention that we will use in the book, which should be noted carefully because the products are in transition from Candle terminology to the new terminology related to IBM Tivoli Monitoring Version 6.1. We conform to IBM Tivoli Monitoring V6.1 terminology as much as possible, although for some direct references to screen shots, we use the terminology that is shown in the figure.

The product is installed using the IBM Configuration Tool. This is explained for new users who want to install IBM Tivoli OMEGAMON XE, because the process is somewhat different from standard IBM products that are installed on z/OS. The concept of runtime environment is explained here as well.

On the operation side of IBM Tivoli OMEGAMON XE, we discuss problem determination and tracing concepts and facilities that are available for IBM Tivoli OMEGAMON XE, as well as performance considerations. As IBM Tivoli OMEGAMON XE is meant to monitor performance of the system, its overhead should not affect overall system performance.

Last but not least, we offer sample usage scenarios of performance management using IBM Tivoli OMEGAMON XE products.

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

SG24-7155-00 ISBN 0738497290