

IBM z Systems

**IBM Monitoring OMEGAMON
Best Practices &
Performance Guidelines V2**

October 2020

**Matt Hunter – IBM
Rocky McMahan – Rocket**



Table of Contents

- 1. Objectives..... 5**
 - 1.1 Document Updates and Revisions..... 5*
- 2. Requirements..... 5**
- 3. IBM OMEGAMON shared documentation 5**
- 4. Maintenance Documentation 5**
- 5. PARMGEN Configuration for OMEGAMON/TMS Products on z/OS..... 6**
- 6. Components and architecture..... 6**
- 7. TEPS..... 8**
 - 7.2 TEPS performance..... 8*
 - 7.3 TEPClnt can affect portal server performance 8*
- 8. TEMS 9**
 - 8.1 Major processes in TEMS..... 9*
 - 8.2 TEMS Hub Platforms 11*
 - 8.3 TEMS overhead isolation procedures..... 12*
 - 8.4 Tuning situations..... 17*
- 9. IBM OMEGAMON on z/OS..... 22**
 - 9.1 Monitoring Real Storage on machines..... 22*
 - 9.2 Migrating sysplex proxy overhead 24*
 - 9.3 USS Monitor..... 25*
 - 9.4 Crypto Monitor..... 25*
 - 9.5 Performance Tips Using the OMEGAMON z/OS 3270 Interface 26*
 - 9.6 Common Storage Area (CSA) Analyzer..... 26*
- 10. OMEGAMON for CICS..... 27**
 - 10.1 Situation synchronization 27*
 - 10.2 Excessive number of situations to create exceptions..... 28*
 - 10.3 Auto-discovery of CICS regions..... 28*
 - 10.4 Situations 30*
 - 10.5 Situation distribution..... 31*
 - 10.6 Historical data collection 31*
 - 10.7 Common collector overhead 32*



IBM z Systems **IBM Monitoring – OMEGAMON Best Practices & Performance Guidelines V2**

10.8	<i>Classic Historical bottleneck data collection</i>	32
11.	OMEGAMON for JVM	33
12.	OMEGAMON for Messaging	33
12.1	<i>KMQ Monitoring Agent & MQ Broker Agent</i>	33
12.2	<i>Getting started with configuration</i>	33
12.3	<i>Granting authorities to IBM MQ Monitoring agent</i>	33
13.	OMEGAMON for DB2	36
13.1	<i>Classic Background data collectors</i>	37
14.	OMEGAMON for IMS	38
14.1	<i>Classic Background data collectors</i>	38
14.2	<i>Controlling auto discovery</i>	38
14.3	<i>Controlling background data collectors</i>	38
14.4	<i>Product-provided and customer situations</i>	39
14.5	<i>Additional consideration for Version 5.x</i>	39
14.6	<i>Application Trace Facility (ATF)</i>	40
15.	OMEGAMON for Storage	41
15.1	<i>Situations</i>	41
15.2	<i>Data Collection</i>	42
16.	OMEGAMON for Mainframe Networks	45
16.1	<i>Processor use for monitoring networks on z/OS systems</i>	45
16.2	<i>Understanding how real-time data is collected</i>	45
16.3	<i>Determining which systems and TCP/IP address spaces to monitor</i>	46
16.4	<i>Determining which types of real-time data to collect</i>	47
16.5	<i>Defining data collection intervals</i>	51
16.6	<i>Defining display intervals</i>	52
16.7	<i>Defining and running situations</i>	53
16.8	<i>How historical data is collected</i>	53
16.9	<i>Determining which types of historical data to collect</i>	55
16.10	<i>Designing workspaces</i>	57
17.	OMEGAMON 3270UI Product Guidelines	59
17.1	<i>OMEGAMON 3270UI</i>	59
17.2	<i>Enhanced 3270 Interface</i>	59



IBM z Systems **IBM Monitoring – OMEGAMON Best Practices & Performance Guidelines V2**

18.	OMEGAMON Presentation Best Practices	63
18.1	<i>TEPs Presentation Best Practices.....</i>	63
18.2	<i>Alerting Best Practices.....</i>	93
19.	Architecture Planning and deployment strategy	121
19.1	<i>Architecture Components - Constraints.....</i>	122
19.2	<i>Deployment Goals.....</i>	122
19.3	<i>Deployment Guidelines.....</i>	122
19.4	<i>Locating and sizing the hub TEMS.....</i>	122
19.5	<i>When to use multiple hub monitoring servers.....</i>	123
19.6	<i>Locating and sizing the remote TEMS.....</i>	123
19.7	<i>Optimizing situations</i>	124
19.8	<i>Locating and sizing the TEPS.....</i>	125
19.9	<i>Memory and disk requirements.....</i>	126
19.10	<i>Architecture Guidelines</i>	129
19.11	<i>High Availability / Fail Over</i>	131
20.	Product Codes.....	132
21.	Notices	136
21.1	<i>Trademarks</i>	137

1. Objectives

The objective of this document is to provide clear and concise tuning guidelines for IBM® OMEGAMON® on z/OS® products. These products include OMEGAMON on z/OS, CICS®, Messaging, IMS™, DB2®, Mainframe Networks and Storage. This document also provides tuning guidelines for the TEMS and TEPS.

1.1 Document Updates and Revisions

This document is a “living” document. It will be updated as new and relevant information is gathered, reviewed, and deemed suitable for placement in this document. You are encouraged to send information, recommendations, and comments to: Rocky McMahan rmcmahan@rocketsoftware.com or Matt Hunter mshunter@us.ibm.com

2. Requirements

In-depth knowledge of the installation, configuration, deployment, and customization of IBM Monitoring software on z/OS and distributed systems is assumed.

3. IBM OMEGAMON shared documentation

Follow this link to the latest Shared Documentation reference material:

https://www.ibm.com/support/knowledgecenter/SSAUBV/com.ibm.omegamon_share.doc_6.3.0.2/shared_welcome/welcome.htm

4. Maintenance Documentation

- z/OS Preventive Maintenance Strategy to Maintain System Availability
http://www-03.ibm.com/systems/resources/zos_preventive_maintenance_strategy.pdf
- Enhanced HOLDDATA
<http://service.boulder.ibm.com/390holddata.html>
- Consolidated Service Test
<https://www-01.ibm.com/support/docview.wss?uid=isg3T1027575>
- Parallel Sysplex - Software Management for Availability
SG24-5451
<http://www.redbooks.ibm.com>
- Red Alerts
<https://www14.software.ibm.com/webapp/set2/sas/f/redAlerts/home.html>

5. PARMGEN Configuration for OMEGAMON/TMS Products on z/OS

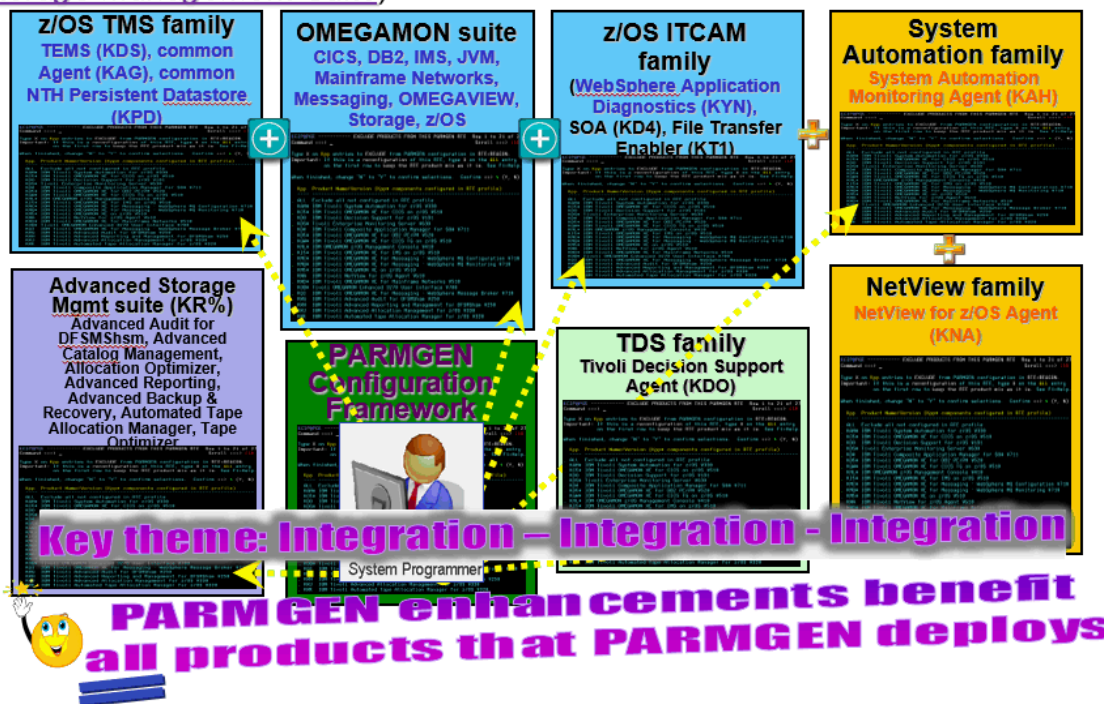
PARMGEN configuration support for configuring the IBM OMEGAMON and other Management Services (TMS) dependent products on z/OS.

Follow this link to the latest PARMGEN reference material:

<http://www.ibm.com/support/docview.wss?uid=swg21417935>

The products listed below, along with their dependent configurable components, are supported to create a brand-new runtime environment (RTE) using the PARMGEN z/OS Configuration Tool. The following versions are listed as the *minimum** versions that PARMGEN mode supports for these products:

z/OS Product Families Supported by PARMGEN (a number of Agents are included in our **OMEGAMON Performance Management Suite**, **IBM Service Management Suite for z/OS** and **Advanced Storage Management Suite**)



6. Components and architecture

This section explains the overall architecture of IBM OMEGAMON. The architecture is based on IBM OMEGAMON V5.x. In this section, we explain the structure and mechanism of IBM OMEGAMON monitoring and the interrelationship and dependencies of its main components.

IBM OMEGAMON products on z/OS consist of several independent components that make up the overall monitoring environment as shown in Figure 3-1.

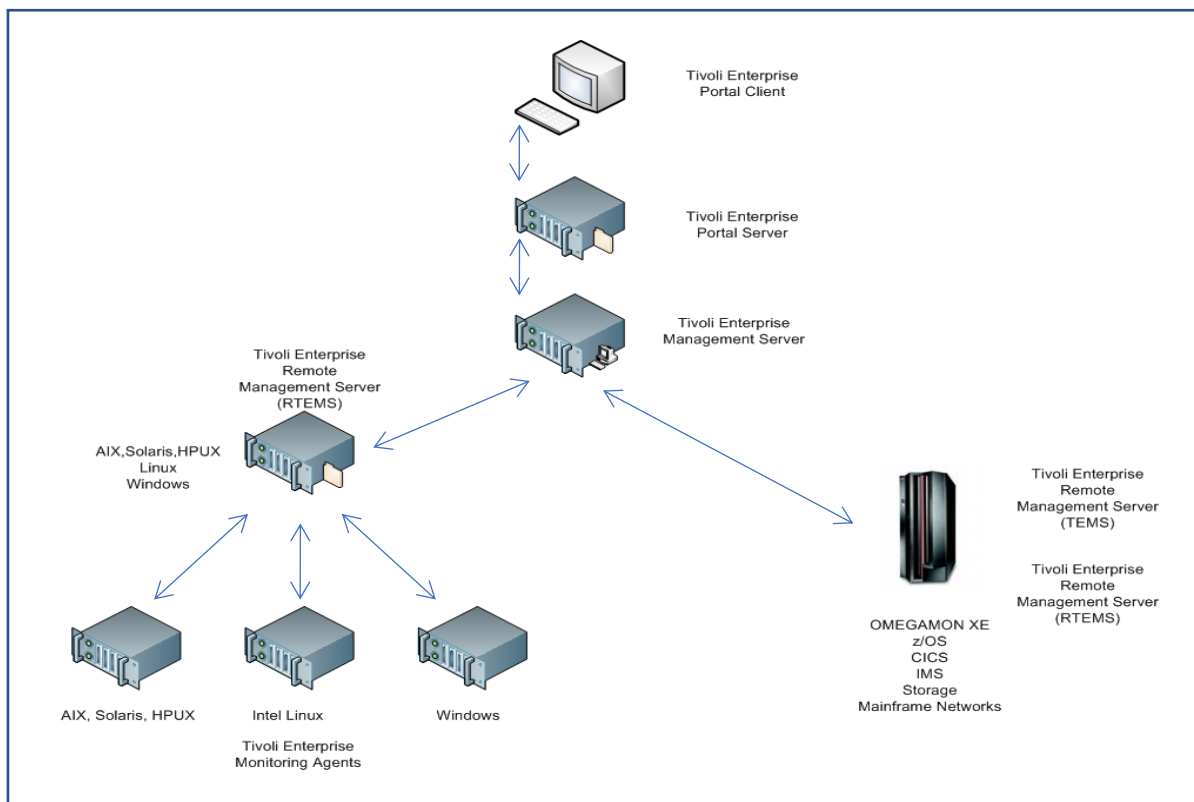


Figure 3-1

As shown in Figure 3-1, these are the components of IBM OMEGAMON:

- TEP, a Java based user interface that communicates with TEPS to request and retrieve monitoring data. Depending on how it is installed, you can view the data using a desktop application (desktop mode) or Web browser (browser mode).
- TEPS, 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 TEMS and can be accessed by the TEPclient.
- TEMS has these functions:
 - Consolidates the data collected by the agent and distributes the data either to the TEPS, or to the Hub TEMS if it is a remote TEMS.
 - Evaluates column function for situations that cannot be resolved at the TEMA.
 - Sends an alert to the TEPS when specified conditions are met and receives commands from the interface and distributes 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 TEMS Enterprise Information Base (EIB) tables when a monitoring agent is installed.
 - Specifies security settings such as a log on password requirement.
- TEMA 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 TEMS. Actions can be performed locally on TEMA. These agents also respond to and process queries from TEPS. These agents can issue commands to the monitored system or application. This document discusses the monitoring agent specifically for the z/OS environment.

7. TEPS

The TEP Server is very similar to that of any application server. Its performance is directly related to the performance of its underlying database, the TEMS. For the monitoring server to be responsive and provide optimal performance it must first be able to respond quickly to requests from the portal server.

The portal server does not provide for great flexibility in tuning its overall performance. Any client/server application is dependent on its network and intervening components. Here are some basic guidelines for getting the best performance from the portal server and ultimately, quick responses at the portal client.

7.1.1 TEMS performance (first)

A poorly performing monitoring server database configuration can only lead to poor performance at the portal server and ultimately at the portal clients:

1. Do not share applications with the monitoring server on the target server unless the monitoring server is running on a z/OS LPAR. When using a distributed monitoring server, the monitoring server should be the only application using that server other than a TEMA.
2. Place the monitoring server on a server/LPAR that provides all the memory that the monitoring server requires and multiple physical processors.
3. Disks with very fast access times can improve monitoring server performance.
4. Separate the monitoring server from the portal server by installing them on different servers. From a z/OS perspective, the portal server should typically reside on a distributed server or z/Linux server. If documentation suggests sizing > 3GB then 64-bit operating system is recommended.

Important: Do not connect TEMAs directly to the Hub TEMS as this can cause unnecessary overhead at the monitoring server (handling agents) and this can ultimately be reflected back to the portal server. Connect the TEMAs to a Remote TEMS (RTEMS) and then connect that RTEMS to the Hub TEMS.

7.2 TEPS performance

If possible, use a server that has multiple, fast disk drives, large memory configuration (4 gigabytes), and multiple processors (2 or more). If documentation suggests sizing > 3GB then 64-bit operating system is recommended.

1. Before installing the portal server, you must first install the DB2 product on one disk drive.
2. Install the portal server on a separate disk drive.

7.3 TEPClient can affect portal server performance

If you log into the portal server with the Physical navigator as your primary view, all the objects load straight into memory. If you create a logical view and drag the objects into folders (for instance: server categorized by application), as you open each object tree only those objects are loaded into memory. Create TEPS few Navigator views as each is loaded into TEPS memory on first reference and not purged until TEPS is restarted. Instead use fewer navigator views and assign users subsets of this navigator view by limiting products.

Use Java Webstart client or override default JAVA cache size 64mb to 256 MB if using browser mode
Portal clients that have low screen refresh intervals and multiple workspaces displayed can cause overhead at the portal server, the monitoring server, and the agents.

8. TEMS

The TEMS, as the main processing component for IBM OMEGAMON products, plays a vital part for the solutions. The hub TEMS acts as the central consolidation point for all monitoring environments in the system. Typically the TEMS is a relatively more processor-intensive system. It is not necessarily memory intensive or I/O intensive, except for the optional persistent data stores (PDSs). This section analyzes the processor behavior of TEMS:

8.1 Major processes in TEMS

These are the processing requirements in TEMS that typically require processing power:

- Situation evaluation
- Situation state changes
- Managing TEMAs
- ASCII / EBCDIC conversions
- Historical data ASCII to EBCDIC conversion
- Historical data storage in monitoring agents
- Data movement and sorting
- TEMS resident data collection
- Historical for resident collection

8.1.1 Situation evaluation

The TEMA evaluates most situations, but some predicate evaluations must be performed on the TEMS, such as column operations. Column operations, such as the counting of occurrences or determining a maximum or minimum value, cannot be performed at the agent. TEMS retrieves data from the agent and performs the column function. This means higher processing requirements at the TEMS. You should try to prevent this processing from occurring at the hub TEMS, by connecting agents w/ these situations to remote TEMS, if possible.

8.1.2 Situation state changes

Typically, situation state changes are a small part of hub TEMS processing. If situation state changes are not part of Hub TEMS processing, it is usually easy to identify and tune. Every time a situation changes state, the following processing occurs in the TEMS:

1. The TEMA informs its local TEMS regarding the state change and can start a policy thread, process imbedded situations, or both.
2. The state change is forwarded to the hub TEMS 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 not valid 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.

If you are using the Managed System Offline situation which always run in the Hub TEMS make sure there are not numerous false positives. Clean up / remove any obsolete systems or ensure the situation predicates cause these obsolete systems not to evaluate as true.

8.1.3 Managing TEMAs

When agents are connected to the hub TEMS, it has to process their heartbeats. With a large number of agents, this can consume measurable processing and slow the responsiveness of the clients. The overhead can be offloaded to a remote TEMS. A remote TEMS also assists in situation distribution through a fan-out process, in which the hub TEMS distributes one situation to the remote TEMS and the Remote TEMS distributes that situation to every one of its agents on the distribution list.

8.1.4 Minimizing ASCII / EBCDIC conversions

Whenever data passes between the z/OS system and distributed computers (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 TEMS, the TEMS must convert ASCII data to EBCDIC for every request. To eliminate the conversion, connect distributed agents to a distributed hub or remote TEMS. The configuration that causes the highest conversion processing is a distributed TEMA connected to a z/OS based hub TEMS connected to TEPS.

Both the hub TEMS and the TEPS must convert the data. Using remote TEMSs, you can offload data conversion processing from the hub TEMS. For example, if your hub TEMS is hosted on a z/OS system, by attaching distributed TEMAs to a z/OS remote TEMS, the ASCII to EBCDIC conversion will occur in the remotes. Because the TEPS 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.

8.1.5 Historical data ASCII-to-EBCDIC conversion

If you store historical data at the agent as recommended, there is no TEMS data conversion. If you collect history at the TEMS where agents of different protocols are connected and it is 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 TEMS to distributed based TEMS.

Important: It is best to collect historical data “at the agent.”

8.1.6 Historical data storage in monitoring agents

Avoid historical data storage at the hub TEMS, if possible. Instead, store historical data at the agent or remote TEMS. If you store history in the PDS (Persistent Data Store) files, it can be converted as explained but it must be written to DASD. This can use cycles to store and retrieve. In addition to steady state cycles, large or multiple PDSs affect TEMS restart time, as they are read completely at startup.

8.2 TEMS Hub Platforms

When we move the hub TEMS from a z/OS system, some of the processing load is offloaded from the z/OS based TEMS and runs 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 TEMS:

- 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, the largest source of hub TEMS processing is often code that must run on a z/OS system. 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 computer, even if the hub is moved to a distributed platform. You can experience higher processing in the hub TEMS than in other z/OS based TEMSs. This can occur for one of the following reasons:

- 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 TEMS or tuned.

- TEMS resident data collectors

- OMEGAMON 3270 CUA for Z/OS component's WLM probe

The OMEGAMON 3270 CUA Z/OS component of IBM OMEGAMON on z/OS WLM probe must remain on an LPAR in a TEMS, which can be a hub or a remote. If OMEGAMON 3270 CUA Z/OS 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 TEMS probes

Any products running within a TEMS such as IBM OMEGAMON on z/OS cannot be moved. You can still have a remote TEMS on the z/OS image.

Higher processing use by these functions on the hub TEMS might be related to the number of subsystems or the volume of their workload.

- Sysplex proxy

If you have IBM OMEGAMON on z/OS, it is likely that the sysplex proxy data consolidation function has defaulted to running in the hub TEMS. This is a high-processing requirement function and can be relocated easily. Simply configure the hub TEMS as not eligible for the sysplex proxy, which will offload cycles from the hub TEMS. This is typically not done because the first TEMS defined is the hub TEMS and, by default, the configuration tool assigns the first TEMS as the sysplex proxy. You need to change this using IBM Configuration Tool after the first remote TEMS has been defined to the hub.

8.3 TEMS overhead isolation procedures

As discussed before, the primary processing in the TEMS is either through user requests or situations firing. Not much can be done for user requests that are not automated. Usually, you want to start all of your required situations automatically to optimize processing with situation synchronization, then isolate the problem using either of the following methods:

- Running IBM OMEGAMON on z/OS PSW sampler INSPECT function on z/OS based TEMS, you can identify the module that consumes processing. Positive identification can be performed after disabling situations related to the modules and measuring the performance again.
Important: The INSPECT function is available from the OMEGAMON 3270 CUA for Z/OS and the TEP. If using INSPECT to monitor the TEMS, use the OMEGAMON 3270 CUA; otherwise, you will add processing requirements to the TEMS to display the metrics that in turn will add more data to the INSPECT results.
- Measuring processing second usage of TEMS and disabling one-by-one the product's situations. This allows for the identification of which solution's situation contributes the largest amount of processing. Note that some of the situation overhead can also be generated by UADVISOR situations. These UADVISOR situations are collected on their own interval. The following sections provide detailed procedures for problem isolation.

8.3.1 Isolating a processing consumption source using the INSPECT function

The following procedure can be used to investigate TEMS processing using the OMEGAMON 3270 CUA component of the IBM OMEGAMON on z/OS INSPECT function. 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.

Important: If you need to investigate TEMS address space processing consumption at this level of detail, it should be done with assistance from an IBM IT Specialist, an IBM Software Services Consultant, or an IBM Software Support Specialist. Additionally, a Problem Management Record (PMR) should be opened so that the issue can be properly tracked and any defects properly documented.

1. With all situations running, use the OMEGAMON 3270 CUA 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. The easiest way is to type INSP NOLIMIT as soon as you start inspect
2. Make sure that CPU issue you are concerned about is occurring, this can be done w/ external CPU measurement, OMEGAMON JOBN &zoom w/ TCP2 minor on the INPECT screen, or by verifying samples used is 5-10% of samples taken. See also 5.4.4, "Tuning Situations".
3. Determine which module has the highest overhead. The module naming standard Kxx should indicate the product or component. See Appendix A, "Product codes" for internal code names. Figure 5-3-1 shows KDFxxxxx modules that are part of the IBM OMEGAMON 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(*) HOT
+ 58 006AFE88 KLV          88.92=====>
+ 54 006D9698 KLV          5.49|
+ 50 006D9338 KLV          5.03|
+ 52 006A52D8 KDFSHSML    .00||

```

Figure 5-3-1 OMEGAMON 3270 CUA Z/OS INSPECT (Classic interface)

3. Disable all the situations associated with the running of that module.
4. Measure the processing of the TEMS without the situations or background data collector started, and determine the percentage reduction.
5. Look at the RKLVLG 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 Section [5.4 Situation Tuning](#) section on Situation synchronization.
6. If you still experience problems, it might be a hidden situation. Determine which UADVISORS are running.

8.3.2 Isolating using CPU seconds

If you determine that a processing 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 3270 CUA INSPECT function, the module names cannot be resolved, or it is running on a distributed platform. You must have all of the situations automatically started to maximize situation synchronization efficiency. If you start one situation at a time, none will share data collection with other situations.

The following procedure is used to dynamically disable, not enable, situations. Remember to measure processing at the TEMA, remote TEMS, and hub TEMS for each of the steps. The following steps provide a high-level overview of the procedure.

Important: If you need to investigate TEMS address space processing consumption to this level of detail, it should be done with assistance from an IBM IT Specialist, an IBM Software Services for IBM Software 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 processing for one minute longer than the highest situation interval to ensure that all background collector executions are captured. This typically takes six minutes.
2. Disable all situations associated with a given table / probe. Measure the process / address space processing 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.
4. Repeat steps 2 and 3 until all of the situations are disabled
5. Measure the process or address space processing for several minutes. Subtract the remaining CPU seconds (not associated with situations) from the CPU seconds number in step 1. This number is total CPU seconds associated with all situations.
6. Convert numbers for each table that were gathered in step 3 to a percentage of the situation CPU seconds 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 RKLVLLOG 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 might be overly complex, having more predicates than needed to determine a problem.
10. If you are still experiencing problems, it might be a hidden situation. Find out which UADVISORs are running, as outlined in the next section

With OMEGAMON DE on z/OS, you can use a custom SQL query against the situation table; it will display the usually hidden UADVISOR situations and their sample interval in REEV_TIME column. Historical UADVISORS all have PDT columns that begin with *IF *HISTRULE nomenclature.

Example: `"SELECT SITNAME, REEV_TIME FROM O4SRV.TSITDESC WHERE PDT LIKE '*IF *HISTRULE*';"`

8.3.3 Disabling products and functions

8.3.3.1 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 OMEGAMON on z/OS.

You might be directed by IBM Software Support to issue these commands. IBM Customer Support will provide the command and syntax that is 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 this, shut down the specified component. If the problem persists, it is not the stopped component.

Example 5-3-3-1 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
```

8.3.3.2 Selective load

You can effectively disable a product or component by selectively not loading the code that provides that product or component.

Important: Modify the load lists for any IBM OMEGAMON 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 TEMS or TEMA, has a KxxSYSIN member in RKAN PAR that lists all products that are to be loaded. To disable an entire product, comment out its LOADLIST statement. For example, the TEMS has a member in RKANPAR(U) called KDSSYSIN, which has entries such as LOADLIST(KM5LLIST). This particular entry loads the IBM OMEGAMON on z/OS sysplex component modules.

To disable the entire product:

Copy RKANPAR(KxxSYSIN), to RKANPARU and edit it, 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:

Copy RKANPAR(KxxLLIST), to RKANPARU and edit it, to comment out the modules in question. Be aware that modules can be in more than one LOADLIST. For example, IBM OMEGAMON on z/OS (KM5) and IBM OMEGAMON for Storage (KS3) both use the shared DASD data collector and therefore will have LOADLIST statements for this component

8.4 Tuning situations

Situation is the mechanism in IBM OMEGAMON 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.

8.4.1 Situation processing

Situations are a mechanism in IBM OMEGAMON 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 TEMS.

The frequency of state changes affects hub and remote TEMS capacity. It also affects storage requirements because these events are saved in memory and written to disk. The sampling intervals themselves affect TEMS capacity.

Several important considerations for situation processing that can greatly affect the performance of the IBM OMEGAMON system are discussed here:

- Situation distribution
- Situation predicate processing
- Situation threshold and interval
- Situation synchronization
- Situation and policies
- Embedded situations

8.4.2 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 TEMS, so all available rows will be sent back to TEMS from the agent, with high network use, memory, and processing requirements as a result. Try to avoid the predicates such as MAX, MIN, and AVG on a column.

8.4.3 Situation threshold and interval

IBM supplies a set of default thresholds with each IBM OMEGAMON solution. However, the defaults might 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, but situations that detect a resource shortage condition probably need a more frequent interval.

Some IBM OMEGAMON products have specific critical threshold situations for stall and crash prevention that automatically start by default, but this might not be true for their warning situations. At many sites the warning yellow alerts 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 automatically started. With IBM OMEGAMON you can set different intervals for every threshold. Be aware that picking different intervals will cause higher overhead. Some products, such as IBM OMEGAMON for CICS on z/OS, have preselected critical threshold situations for stall and crash prevention automatically started as a default. However, they do not auto-start any warning situations.

The predefined situations might 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 might not be critical 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 can be different for various applications running under the same subsystem.

IBM OMEGAMON attempts to have low overhead defaults. However, no vendor can know your environment or what is relevant to your business applications. IBM OMEGAMON was designed to provide in-depth analysis, because you need these capabilities to deal with problems when they are 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 OMEGAMON z/OS uses the same default thresholds as the associated OMEGAMON 3270 CUA 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 OMEGAMON product to run KXEMIGR.

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 can have more than one way to detect the same problem and not all situations have the same processing requirement. IBM OMEGAMON provides Boolean logic and multiple ways to detect complex

situations. For example, a CICS transaction rate of zero with high processing 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.

8.4.4 Situation synchronization

One of the biggest savings in situation processing is to perform situation synchronization, sometimes 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 situation requirements must be fulfilled, as shown in Table 5-4-4.

Table 5-4-4 Synchronization eligibility requirements

Situation definition	Needed
Same attribute group required	Yes
Same time interval required	Yes
Can be restarted	Yes
Must be automatically started (except UADVISOR cannot have auto-start set to SYN)	Yes
New / update situation supported (requires TEMS 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	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 can 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 the RKDSRULD file. The RKDSRULD file contains the situation synchronization rules. The actual limitation is the size of the rule. Typically, 8 - 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 TEMAs can automatically collect information once for multiple automatically started situations with the same interval.

To determine whether a situation has been included in synchronization, you can browse the RKDSRULD dataset on the TEMS that the TEMA is connected to.

Any duperized situations are located at the beginning of the file, with names that start with `_Z_`. Looking at the list of synchronized situations, you might see several `_Z_` entries with different suffixes. 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 for message, KRAIRA000. This indicates that a synchronized (duperized) situations are being started.

8.4.5 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 TEPS 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 is additional scheduling of the collector. Depending on the level of the code, the resulting behavior is that in IBM Monitoring V6.1 (and after), the original situation is not eligible for synchronization but shares the collected data with the policy. This offers some processing reduction compared to previous levels.

8.4.6 Embedded situations

Embedded situations are situations that 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 independently 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 automatically started).

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, the Weekday situation runs 21 times if Weekday is automatically started. This can mean unnecessary processing.
- An 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. 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. You can use a more meaningful name for Weekday, such as CICS_Weekday_Alerts.

Any time you use embedded situations, neither the parent nor child are eligible for situation synchronization. If they were, it could 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. By doing this, all situations are eligible for situation synchronization and the TEMS situation monitor does not have to process (filter out) alerts during the time frame when they are not applicable. Having all situations synchronized reduces the total number of situations, which also reduces TEMS and agent overhead.

These steps outline the procedure for coding time policies to reduce the agent and TEMS processing requirements:

1. Create two situations that check the day of the week.
 - Weekday situation: check for weekday $\text{DAYOFWEEK} \geq 2$ and $\text{DAYOFWEEK} \leq 6$
 - Weekend situation: checking for weekend $\text{DAYOFWEEK} = 1$ or $\text{DAYOFWEEK} = 7$
2. Create two policies:

- Weekday_Policy that, based on the active date situation (Weekday), restarts all the situations that should be active.
- Weekend_Policy policy that, based on the inactive period, stops all the situations that you started in the first policy.

By default, policies are set to auto-start. 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 are not eligible for situation synchronization processing unless they are automatically started.

The policy executes when situation Weekday_Starter_Situation is true. At that time, the policy starts four situations.

3. Make sure that all product situations are automatically started so they will become synchronized. You do not need to start the time/date dependent situations, because that is 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 typically generated.

9. IBM OMEGAMON on z/OS

For some enterprises, the operating system 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 you can often eliminate through the IBM OMEGAMON architecture. The main focus of this section is how to avoid duplicate data collection using techniques unique to IBM OMEGAMON on z/OS.

9.1 Monitoring Real Storage on machines

Monitoring Real Storage on machines with large amounts of real storage can be quite expensive. There are four product provided situations that reference this table.

- OS390_Available_Frames_Crit
- OS390_Available_Frames_Warn
- OS390_Frames_Online_Crit
- OS390_Frames_Online_Warn

Highly recommend that you ensure the remaining situations are situation synchronized and that one _Z_ situation is running instead of these 4. Also use the least frequent sampling interval that makes sense for your environment.

- Unnecessary situation samples

You do not have to enable some other situations, such as expanded storage exceptions, because there are no mechanisms to influence them in the current new zSeries processors. These are the expanded storage situations that should not be enabled:

- 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

Monitoring Real Storage on machines with large amounts of real storage can be quite expensive. There are six product provided situations that reference this table. The last two UIC situations can be replaced by you with situations referencing a different table, change Real_Storage.Unreferenced_Interval_Count to System_Paging_Activity.Unreferenced_Interval_Count. This will be updated in a future release of OMEGAMON on z/OS when you reseed your situations.

- OS390_Available_Frames_Crit
- OS390_Available_Frames_Warn
- OS390_Frames_Online_Crit
- OS390_Frames_Online_Warn
- OS390_Unref_Interval_Cnt_Crit
- OS390_Unref_Interval_Cnt_Warn

Highly recommend that you ensure the remaining situations are situation synchronized and that one _Z_ situation is running instead of these 4. Also use the least frequent sampling interval that makes sense for your environment.

9.2 Migrating sysplex proxy overhead

You do not want the sysplex proxy to run in the hub TEMS, which is busy performing many functions for the TEPS, the remote TEMSs, and potential locally attached agents. But the first TEMS defined to configuration tool is the hub TEMS and, by default, the configuration tool assigns the first TEMS as the sysplex proxy. So, configure the hub TEMS to not be eligible for the sysplex proxy and this will offload cycles from the hub TEMS. You can change this using the configuration tool after the first remote TEMS has been defined to the hub TEMS.

- Reusing or recollecting data

The IBM OMEGAMON on 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 TEPS by viewing sysplex workspaces without recollecting 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 can disable the shared DASD mover. If you do not also have OMEGAMON for Storage you can even disable the background collector. Given the interaction between OMEGAMON on z/OS and OMEGAMON for Storage solutions (and potential maintenance impact), you should contact the IBM Customer Support if you want to pursue this option.

Technote: 1272833

The DASD Z/OS Devices workspace can take a long time to refresh and use high processing when there are many online DASD devices. To speed up the refresh of the DASD Z/OS Devices workspace, and reduce processing consumption in the TEMS, you can specify filter criteria in the queries that are used by this workspace. That reduces the number of devices selected for processing and display at the TEP.

To do this, use the following procedure:

1. Logon to OMEGAMON on z/OS (TEP) and navigate to the Query Editor (Ctrl+Q).
2. Expand the Z/OS System, and then DASD Z/OS Devices entries and click the DASD Z/OS Devices query.
3. Click the **Create Another Query** icon ().
4. Enter a name for the query, for example DASD Z/OS Devices Filtered.
5. Specify filter criteria for the Response column, the Percent Busy column, or both.
6. **Important:** While filter criteria can be specified for other columns, these are the only attributes that will have a noticeable effect on refresh time and processing consumption.
7. Save the query.

8. Click **Edit > Administer Users**, and click your **user ID**.
9. Scroll to the bottom of the Permissions window and click **Workspace Administration**.
10. Make sure that both Workspace Administration Mode and Workspace Author Mode are selected.
11. Click **Apply > OK**.
12. Select the DASD Z/OS Devices workspace, right-click the table display and select **Properties**.
13. Click the **Click here to assign a query**, button and select your new filter query. Then click **Apply**.
14. Repeat step 12 for each of the Bar Chart Views.
15. Click **File > Save Workspace As** and enter a new name for the workspace, for example DASD Z/OS Devices Filtered. Click the button to save the workspace as the default for this Navigator Item.
16. Repeat steps 8 - 10 to reset your permissions to their previous settings.

9.3 USS Monitor

The OMEGAMON USS Monitor (part of OMEGAMON z/OS, running in the TEMS) can be a very expensive component to run in a large system where USS is heavily used. The following parameters documented is in the PTF that introduced them and in the IBM LevelSet PTF that superseded it (UA14515). They can be placed in RKANPARU(KDSENV). The value shown is the default.

KOE_MFSB_WUI=180

KOE_MFSB_TBI=300

KOE_MFSB_MDI=600

Collect File System information every WUI seconds until it is not used for TBI seconds then revert to collecting every MDI seconds.

KOE_THREAD_INTERVAL=6

Collect thread / process / address space information every 'x' seconds. Minimum is 5, maximum is 30.

This one is not documented anywhere. You can use it to turn off ENQ monitoring of the mounted file systems. Without this parameter, OE will do two GQSCAN (GRS call) for each mounted file system at every MSFB interval (above). This could become very expensive in a large sysplex and/or when hundreds of file systems are present.

KOE_MFSB_NOENQD=1

=1 means disable ENQ monitoring. To re-enable it, comment out the parameter.

9.4 Crypto Monitor

Disable the Crypto Monitoring agent product-provided situations.

9.5 Performance Tips Using the OMEGAMON z/OS 3270 Interface

- Use the .VTM command to find sessions that are using a very low refresh auto refresh rate.
- Use the ASG command to specify maximum thresholds for the WAIT and/or TPWT exceptions for long running tasks that wait often. The WAIT and TPWT exceptions can cause high overhead, when trying to determine the reason for the wait. The code will scan every device to see if there is an i/o wait involved. By excluding tasks that normally have long waits, you can bypass the unnecessary ucb scan.
- Use the ASG command to specify maximum thresholds for the FXFR exception for tasks that normally exceed the FXFR threshold. The FXFR exception processes pfte chains for address spaces. This can generate high overhead for tasks that use a lot of storage. Turn off the exception or specify a list of tasks that should be excluded to bypass this overhead.

9.6 Common Storage Area (CSA) Analyzer

- The overhead is based on CSA activity, not the level of use by OMEGAMON users. There is no significant CPU, unless there is a high volume of CSA GETMAINS. This typically occurs on VTAM owning machines or those with high volume IMS control regions. Sample Interval. CPU can be reduced by coding SAMP=99 in the OMIICSA startup procedure. A 1% change in total allocation has to occur before it run the chains. It defaults to 100 or 0% change, which run the chains every second.

10. OMEGAMON for CICS

Many environments 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 OMEGAMON for CICS on z/OS is installed on a z/OS partition.

This section examines the following issues with OMEGAMON for CICS on z/OS:

- Situation synchronization
- Excessive number of situations to create exceptions
 - Auto-discovery of CICS regions
- Situation distribution
- Historical data collection

10.1 Situation synchronization

To reduce OMEGAMON for CICS on z/OS processing requirements and make it comparable to the OMEGAMON 3270 CUA CICS component, the default situation intervals match the OMEGAMON 3270 alert frequencies. By changing the situation intervals to a common value, you can reduce the frequency of data collection by making the situations eligible for situation synchronization.

Table 7-1 lists situations whose intervals should be reduced to 3 minutes, which would match another set of situations that are being collected from the same table.

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

Table 7-1 CICS Region Overview situations

10.2 Excessive number of situations to create exceptions

OMEGAMON for CICS on z/OS distributes situations so that you can duplicate any of the exceptions that were being monitored by the OMEGAMON 3270 CUA.

Most of these situations are not automatically started. 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 3270 CUA CICS but you were asked to assign Warning or Critical status for these conditions. In OMEGAMON 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 as 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 is additional processing. You can 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 then want to ensure that the original 48 situations (three critical and three warning situations per each of 8 LSR pools for a total of 48 situations) were not started or distributed to CICS regions.

10.3 Auto-discovery of CICS regions

OMEGAMON 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 might not want to monitor all the regions. In addition, the more members in the distribution list, the longer it can take to initialize the TEMS for situation processing.

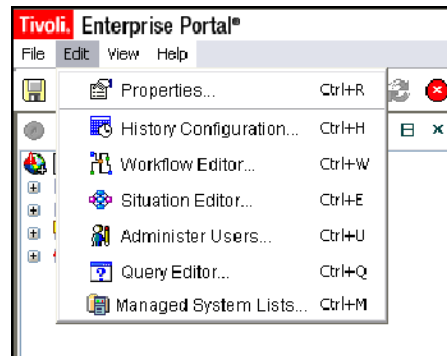
OMEGAMON for CICS on z/OS can be configured to discover subsystems selectively, which eliminates this issue. This can also eliminate the ability to view data from these unmonitored systems at the TEP. This might be preferable because it also keeps them from cluttering the Physical Navigator view.

You can configure production regions to a production hub TEMS and TEPS, 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 that you want to monitor, such as CICS_Production.

The function to edit Managed System Lists is available from multiple start points, such as the Edit option on the Action Bar (Figure 7-3-1). From the Edit Manage System List pop-up window you can create a new MSL or create a copy of an existing MSL.

Figure 7-3-1 Edit Manage System Lists

IBM z Systems **IBM Monitoring – OMEGAMON Best Practices & Performance Guidelines V2**



The CICS_Production Manage System List (see Figure 7-3-2) 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.

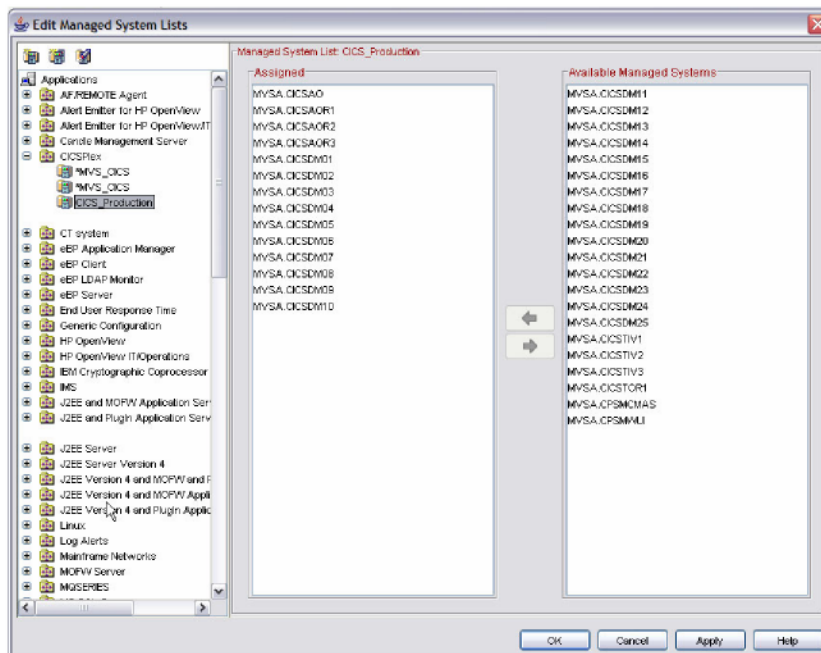


Figure 7-3-2 CICS_Production Manage System List

Similar to the use of the RKC2XMnn DD DUMMY statement for OMEGAMON 3270 CUA CICS to indicate which of multiple KOCCI collector address spaces in one LPAR will manage a CICS region, OMEGAMON for CICS on z/OS uses the RKCPXMnn DD statement to indicate which TEMA 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 TEMA started task should also include the same DD statement to indicate the connection.

10.4 Situations

The following list represents a starter list of “Situations” that should first be considered when implementing the IBM solution for monitoring a CICSplex environment. The subsequent list of product situations is followed by appropriate comments:

- **CICSplex_AtMaxtask_Critical**
Being at or over MAXT is usually a critical condition which indicates a region overwhelmed or under severe stress.
- **CICSplex_DB2Abort_Critical**
For regions accessing a DB2 subsystem, this indicates problems with throughput on a DB2 subsystem and would certainly lead to a processing slow down.
- **CICSplex_EDSADSASOS_Critical**
This is a combination of two product-provided situations indicating a SOS (Short On Storage) condition in either DSA (dynamic storage area) or EDSA (extended dynamic storage area). Seeing this condition is important, as it can cause a region outage, but differentiating between extended and non-extended can be done after receipt of the alert. Combining the situations saves the analysis of two situations rather than one.
- **CICSplex_LSRPoolWait_Critical**
Another combination of 8 product-provided situations. Good alert condition, but we can differentiate which pool after the alert is detected.
- **CICSplex_OSCORLow_Critical**
This is very important. After the condition occurs, there might not be enough room left for opening a file. The region may be in jeopardy of an Sx78 or S80Aabend.
- **CICSplex_Performance_Index_Warning**
This is one of the most important CICS Alerts. This indicates that your response time is not meeting a SLA. This can allow a customer to create a proactive “Yellow alert.”
- **CICSplex_Performance_Index_Critical**
This is another example of an important CICS Alert. This indicates that response time has exceeded a defined SLA by a great deal, thus generating a “Red alert.”
- **CICSplex_ShuntedUOWs_Critical**
Indicates that a complex of related CICS regions is experiencing a slowdown.
- **CICSplex_StorVio_Critical**
Indicates that an application has overlaid storage that it did not own, usually resulting in a region failure or the failure of an application.
- **CICSplex_TakingDump_Critical**
We have a Region which is performing a system dump. The region would be frozen while this occurs.
- **CICSplex_TODUpdate_Critical**
This usually indicates that the region is “hung.”
- **CICSplex_TranDumps_Critical**
A Region has individual transactions taking failures.

10.5 Situation distribution

There are more than 200 product-provided situations (PPSs) for OMEGAMON 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 auto-start, they are distributed to the agent for use by the CICS regions in the event that they are started. This can increase the startup time of the hub TEMS and consumes additional processing.

Review the situations to determine which ones you want to auto-start and remove the *CICS from the distribution lists of any situations you do not want to start automatically or dynamically. IBM Monitor distributes the situations so they are displayed on the TEPNavigator tree associated with the correct Navigator Item (see Figure 7-6). However, many of the situations seen in Figure 7-6 are not automatically started and might not be required in your environment.

When you un-distribute the situations, they are no longer displayed 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.

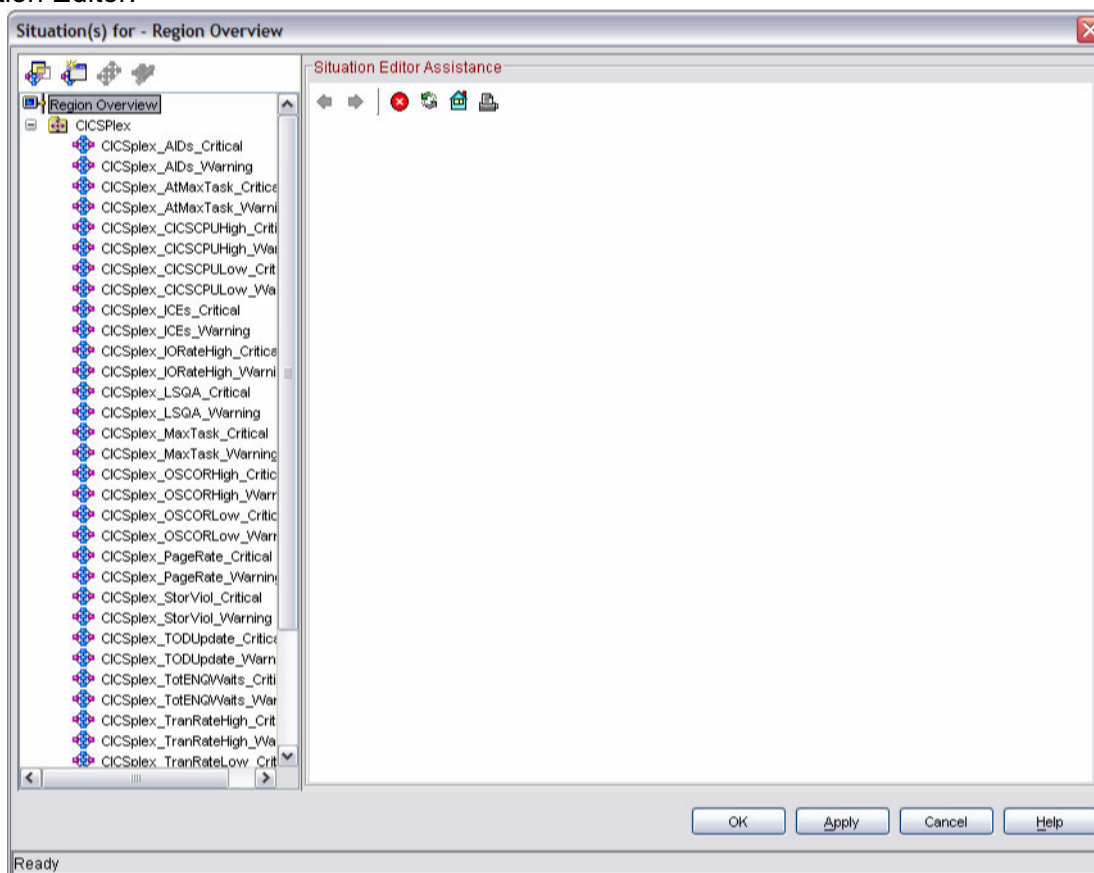


Figure 7-6 Situations associated with (Region Overview) object

10.6 Historical data collection

If your enterprise previously used the CandleNet Command Center (CCC) for CICS, you might have three large Persistent Data Store VSAM files to hold weeks of history information. With OMEGAMON historical data collection and the TDW, you might not need to keep these VSAM files at the original size.

The Persistent Data Stores files need hold only 24 hours of data, at most. This reduces TEMS and TEMA 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 TEMS or TEMA to start. Note: With 3 PDS datasets, one is always a spare and 33% of your DASD is wasted. With 6 PDS datasets, each ½ the size of the original PDS datasets, twice the data can be stored using the same amount of DASD.

10.7 Common collector overhead

There are two primary sources of common collector overhead; 3270 sessions and historical SMF110 data capture for online data viewing (ONDV). Bottleneck analysis and Response Time analysis create very little overhead relative to production CICS they are monitoring. The only tuning for ONDV is to reduce the length of the SMF110 record and the OMEGAMON appendage. The major issue with 3270 sessions is with customers with old CUA profiles with DASD collection enabled or classic users that have OMEGAVIEW driving the sessions w/ DASD enabled. The INSPECT procedure can be used against the common collector, there are TCBS connected for each VTAM session, and for each component activated in every CICS region ONDV (BGR), RTA and Bottleneck (DEX). Be aware that if there are no transactions (test system) that ONDV will be low and bottleneck will show as a higher % of the total, but the total CPU should be low. See technote [#1964210](http://www.ibm.com/support/docview.wss?uid=swg21964210) or just point to <http://www-01.ibm.com/support/docview.wss?uid=swg21964210>

10.8 Classic Historical bottleneck data collection

The Interval Recorder (INTR) does not automatically start by default. This function hardens Response Time Analysis (RTA) and Bottleneck data to SMF at a user defined interval. If both RTA and bottleneck analysis are auto started this function can provide value with little additional overhead as long as the KC2GLBxx section for <Interval_Collector> has Number_DASD_Devices=0. The default is 50 and causes RMF data to be collected every minute which is very expensive. Recommend coding 0 for anyone who is using INTR also known as the Problem Reporter.

11. OMEGAMON for JVM

Enterprise Java Monitoring on z/OS with OMEGAMON - A Practical Guide to Managing JVM Performance on z/OS
<https://developer.ibm.com/mainframe/2019/02/22/find-out-whats-new-in-ibm-z-omegamon-for-jvm-v5-4-0-0-apar-0a56263/>

12. OMEGAMON for Messaging

Below is a high-level summary of the key observations and recommendations found during the IBM OMEGAMON for Messaging V 7.0 measurements. Additional information on these results can be found in the OMEGAMON for Messaging V 7.0 Performance Measurement Report.

12.1 *KMQ Monitoring Agent & MQ Broker Agent*

12.2 Getting started with configuration

Use defaults for parameters to both agents until you understand what else you want to see or change. Both agents, but especially the IBM Integration Bus Monitory agent, require enablement in the monitored product to be able to provide data.

https://www.ibm.com/support/knowledgecenter/SSRLD6_7.5.0/kmq_userguide/kmq_checklist_intro.html

12.3 Granting authorities to IBM MQ Monitoring agent

https://www.ibm.com/support/knowledgecenter/SSRLD6_7.5.0/zos_configuide/tsk-grantauth-kmq-kmc.html

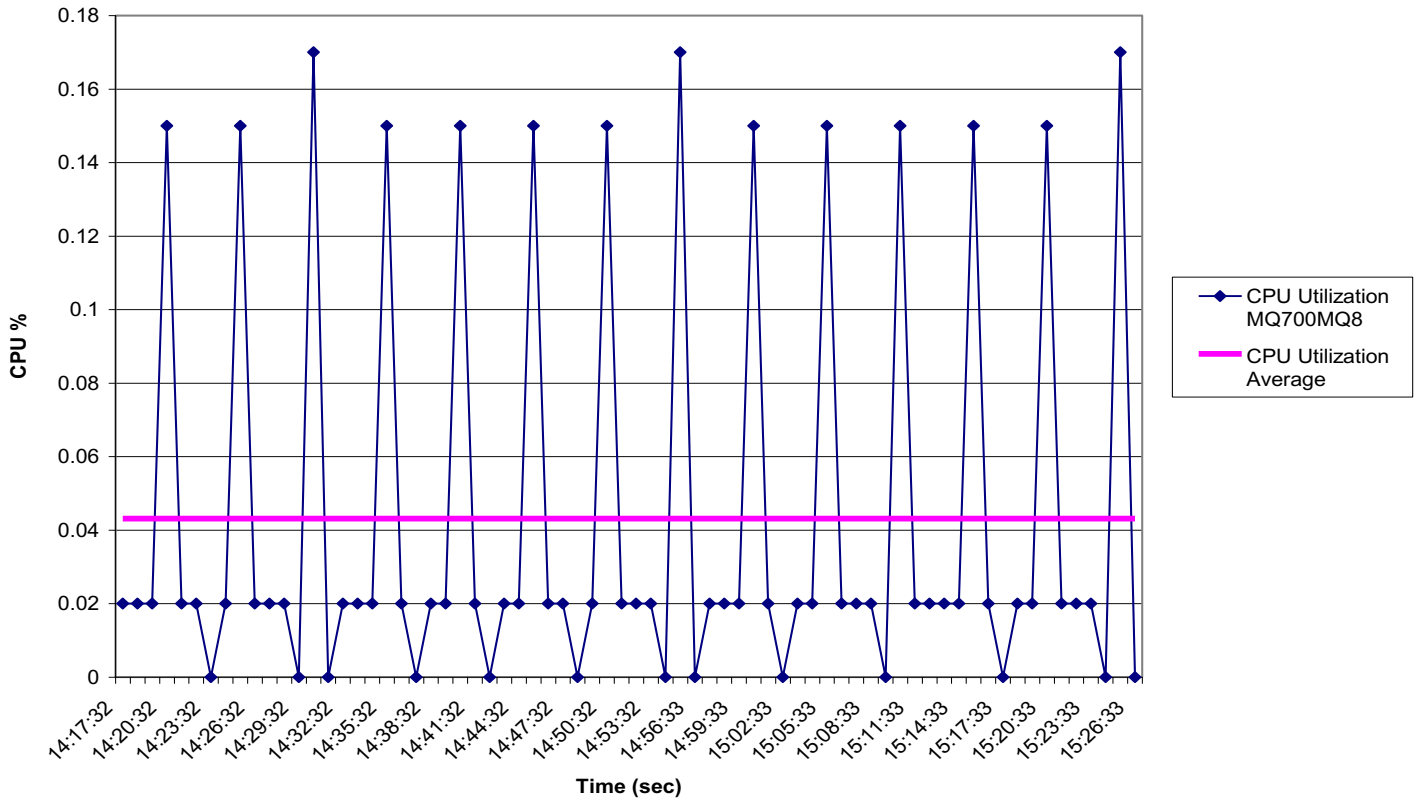
12.3.1 Measurement Results

- All measurements were within the target goals.
- CPU Utilization for both agent and QMgr were both very low, averaging less than .05%.
- Response times were less than 3 seconds.

12.3.2 KMQ - Monitor Agent. 2 Queue Managers, each with 500 objects, 100 channels (20 active) on z/OS.

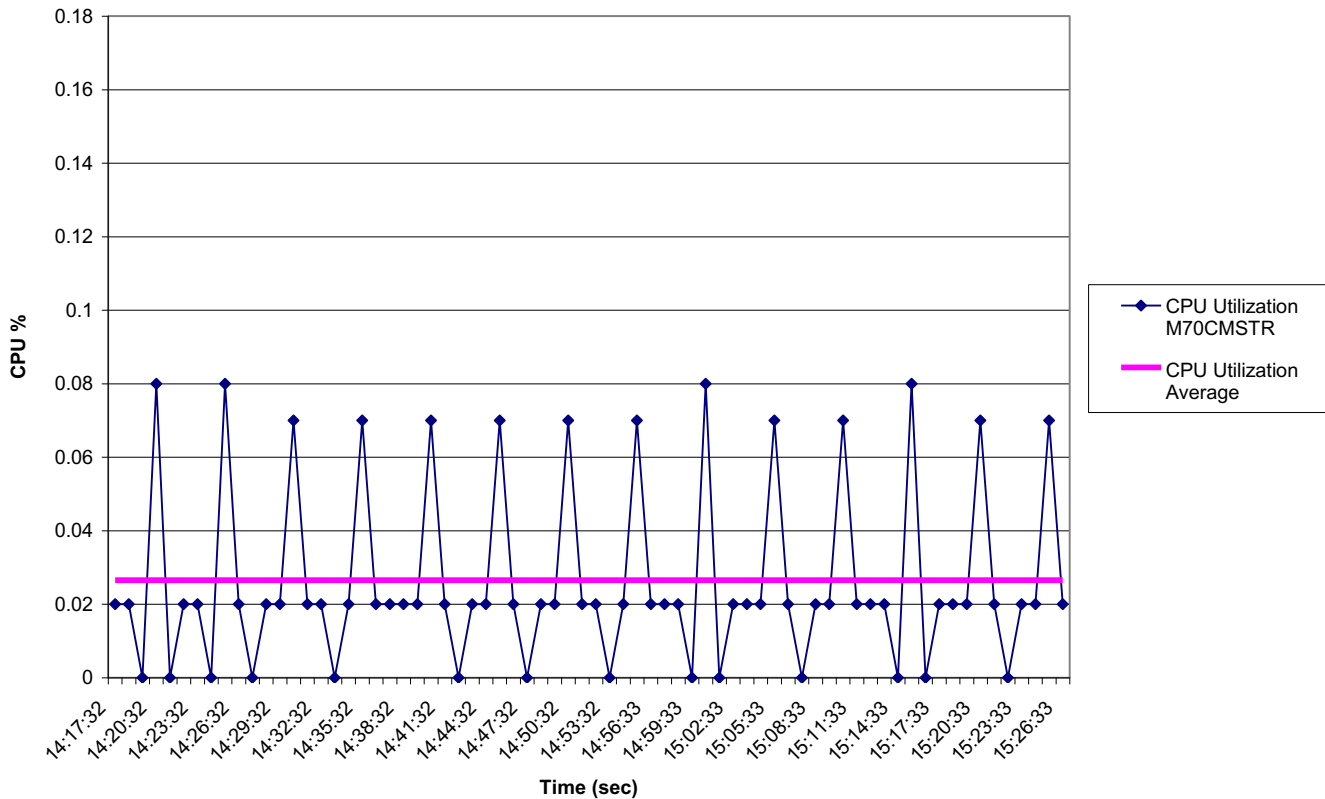
12.3.2.1 CPU Utilization

CPU % Utilization KMQ Agent



Average KMQ Monitoring agent CPU utilization < .05% of a single processor (averaged over 60 minutes), well within the target goal of less than or equal to 5%.

CPU % Utilization Queue Manager



Average Queue Manager Overhead (managed system) CPU utilization < .03% of a single processor (averaged over 60 minutes), well within the target goal of less than or equal to 3%.

12.3.3 Deployment Recommendations

- Existing clients (upgrading) should consider making sure their agent parameters use newer default settings that help improve storage usage and processing overhead:
 - RETAINHIST(120) instead of RETAINHIST(1440) on SET GROUP or SET MANAGER keeps around less data for objects that have been deleted from the queue manager.
 - QDEFTYPE(PREDEFINED) on SET QUEUE avoids increasing the amount of queue data sampled and maintained since the data won't be collected for temporary queues.
 - HISTORY(NO) on PERFORM STARTMON should only be set to HISTORY(YES) if history collection is configured ON for some KMQ tables in TEP; otherwise it wastes storage and data collection overhead for data that is never used.
 - SAMPINT(300) on PERFORM STARTMON sets the sample interval to 5 minutes. The shorter the interval here, the more sampling overhead. With a large number of objects, the difference can be significant.
- Every client should consider the following agent parameters to help improve storage usage and processing overhead:

12.3.4 Deployment Recommendations

- If only interested in performance statistics workspaces, and not message flow definitional details and topology, consider setting agent parameter defaultCollectNodeData to "No". "No" is now the default setting actually because depending on the numbers of message flows, setting this to "Yes" can elongate data collection and increase memory utilization of the agent drastically.
- Do not change the discoveryInterval parameter from the default of 86400 (24 hours) on z/OS systems.
- Do not use the CandleMonitor node. It has been superseded by the data provided with the Message Flow Accounting Statistics feature. It is much more user-friendly and simple to use the accounting statistics data instead of the CandleMonitor node since you will not have to alter message flows. The IBM Integration Bus team does not support you if the CandleMonitor node is enabled in your message flows, and you will be required to disable the node prior to that team working on problems. The node has remained in the product to allow customers time to remove it from Message Flows, but it will no longer be shipped with the product after v750.
- If history collection is not configured nor planned to be configured for Accounting Message Flow Statistics and related "Accounting" tables, set the defaultHistoricalAccountingType to "None" to avoid related processing that can occur for maintaining the data for historical consumption.
- Do not set defaultHistoricalAccountingType to anything other than "Archive" or "None" - the "Snapshot" and "All" settings cannot be used due to the overwhelming amount of historical data produced.
- Historical collections for the 4 tables with names beginning with the word "Accounting" must have the same exact interval defined, such as 15 minutes. However, the only recommended table of these for historical collection is Accounting Message Flow Statistics. The others can produce large amounts of historical data.
- Remember above all that to have the most useful data from this agent, you must enable the broker for archive accounting statistics and for resource statistics. This enablement and other recommendations are covered in the link given for Getting Started with Configuration above.

13. OMEGAMON for DB2

- Sampling intervals

The probes that collect the data for the various report containers can be called 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 does not start.

- Disabling collectors

Collection can be disabled temporarily without recycling the TEMS 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.

- Commands to stop the DB2 collector

- DB2 object analysis overhead

If you have a lot of DB2 objects and Object analysis is enabled in the OMEGAMON 3270 CUA DB2 component, you might see high overhead in the KFACOM module. This module creates indexes in core databases created from data that is moved from the remote TEMSs to the hub TEMS using XCF. You might also see high storage use in extended storage subpool size (145-160).

13.1 Classic Background data collectors

Near Term history collection options were controlled by the start-up parameters in RKD2PAR(COPTxxxx), where xxxx is the DB2 SSID. This member controls the majority of the common collector CPU. The largest CPU consumer with the least value is LOCKSUSP(YES), this now defaults to NO, but customers migrate old options. Lock Conflict is valuable, defaults to on and is not that expensive. The next largest is NEGSQL which also is not worth what it cost to collect. The only expensive option that is worth running is Dynamic SQL (DYNSQL) but some customer get by with DB2/PM Snapshot of the SQL Cache at 5 minute intervals, this captures all highly-used SQL and all long running > 5 minutes. The overhead of these is has a direct relationship to the volume of records collected, this can be checked in classic DB2 w/ the fastpath command H.C.B. Note that DYNSQL collection will have different impact on different DB2 SSIDs based on the applications level of DYNSQL usage. So, you need to type H.C.B on all DB2 SSIDs that are connected to common interfaces where you are concerned about the CPU.

14. OMEGAMON for IMS

This section presents some techniques for managing IBM OMEGAMON for IMS on z/OS.

14.1 Classic 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 OMEGAMON reports. This is done by coding NORESC(DEV) in RKANPAR(KEIOPtxx). If you have thousands of volumes, this could be more than 50 percent of the classic collector's cycles.

Bottleneck Analysis is not that expensive, but its overhead can be reduced by increasing the sample interval from 0.5 seconds. Response time analysis is not expensive at current maintenance levels, but there were some issues corrected in 2010. The biggest expense in OMEGAMON and to IMS subsystem is enabling Application Trace Facility (ATF) in production, this is not recommended. If you do use this there are Performance PTFs to the Journal Logging Facility(JLF) and to the DB2 collection component in 2010. JLF was removed in v530 in favor of z/OS System Logger Facility.

The z/OS System logs can be allocated on DASD or through XCF configuration.

14.2 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. You can use the parameters 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 are not displayed on the TEPnavigator tree. ~~If for some reason, you installed and configured the product before you were ready to use it, you can disable auto-discovery by commenting out the RKANCMD member KIPSTART member's parameter IPDC AUTODISCOVER.~~

14.3 Controlling background data collectors

Disabling auto-discovery of control regions prevents the regions from being displayed on the TEPnavigator tree.

V530 or earlier

For regions that must be available for reactive monitoring, you can selectively disable background collectors with the same RKAN PAR member KIPCNFG as Example 10-1 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 does not start. These parameters are also configured using the IBM Configuration Tool.

Example 10-1 IMS KIPCNFG values

COUPFAC INTERVAL(30) IRLM(BRLM)

MSC INTERVAL(30) IMSID(I51B)

RIAGRPINTERVAL(30) IMSID(I51B)

RIATTEM INTERVAL(30) IMSID(I51B)

V550 or later

Individual collectors may no longer be configured. They are either all ON or all OFF.

Collection can be disabled temporarily without recycling TEMA to aid in problem or overhead determination. The modify commands shown in Example 10-2 stop and restart all of the collectors. If you modify KIPCENFG, this also make the changes effective.

Example 10-2 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)
```

14.4 Product-provided and customer situations

If any product-provided situations are distributed to *IMS, they 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 want to monitor.

Do not start and distribute situations that you will not react to or maintain. When situations generate false alerts, they cause wasted time in investigation and loss of faith in the validity of the alerts. They also can cause additional overhead in data collection, and they certainly waste cycles in evaluation of criteria that are unimportant.

14.5 Additional consideration for Version 5.x

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 TEP, you were looking at less current data.

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 TEP 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 V5.x, not using *IMS and using MSLs for production or test can reduce data collection overhead.

14.6 Application Trace Facility (ATF)

ATF can cause a lot of overhead. Prior to v530, some customers were using TRF as a charge-back system. We removed TRF in v530 and ATF was enhanced to provide functions TRF previously delivered.

To collect chargeback data, the customer must run ATF in production. You may also log the summary records to the IMS log where IMS/PA was enhanced to report on them. Use the LOG=ONLY option as the most efficient to collect chargeback data. No detail records will be collected.

ATF has new controls that allow you to do summary-only collection which avoids all of the detail data collection where most of the overhead occurs. We recommend also to record only exception summary records in production. You can always define a new trace to drill into a specific problem, capture the trace, and turn it off again to limit overhead.

15. OMEGAMON for Storage

This component runs in the TEMS MS address space and the IBM Subsystem address space. It can cause significant overhead if not configured carefully in an environment with 10 000 or more DASD devices. After it is installed, it is best to configure for Storage and adjust the parameters to reduce overhead to an acceptable level.

15.1 Situations

Situations are queries that are run on a specified interval and report their results to the TEMS Situation Monitor. The Situation Monitor then decides whether to forward a situation alert to the TEPS. Because these queries run on a regularly scheduled interval rather than on demand, they can be a source of processing overhead. Here are things you can do to reduce overhead caused by situations.

15.1.1 Situation Distribution

- Review product-provided situations to be sure you are only running the ones that are critical to you.
- Distribute the following situations to only one z/OS image in a shared DASD environment:
 - All situations with space attributes in the predicate
 - Group level, i.e., SMS storage group or user DASD group
 - Data Set Groups
 - Data Set Attribute Database
 - All situations with cache status or cache performance attributes in the predicate
 - Cache hit percent
 - Cache destaging rate
 - All situations related to the hardware
 - Raid rank activity
 - Extent pool activity
 - VTS activity
- RLS attribute group situations should only be distributed RLS attribute group situations should only be distributed to a single system in the sysplex. These attribute groups contain a row of data for the entire sysplex and for each system in the sysplex. All of the data is available from any system in the sysplex (the storage agent need not be active on every z/OS image in the sysplex for this to be true). These attribute groups are:
 - S3_RLS_Dataset_Details
 - S3_RLS_Dataset_Group_Details
 - S3_RLS_Dataset_Group_Summary
 - S3_RLS_Lock_Structure

- S3_RLS_Lock_Structure_CF_Detail
- S3_RLS_Buffer_Pool_Detail
- S3_RLS_Buffer_LSU_Summary
- S3_RLS_Performance_Overview
- S3_RLS_Storage_Class

15.1.2 Situation Take Action

- Take Action on volume or data set situations can cause unexpected results:
 - Situation can be true a number of times, resulting in undesirable results
 - Example:
 - Situation for fragmentation index against all volumes.
 - Add take action to kick off a started task to defrag the volume.
 - If the situation is true for 500 volumes, then 500 tasks can be started. This is probably not a result that you want.
- Recommendations:
 - Split the DASD farm into multiple situations.
 - Time slice the situations.
 - Know your environment.

15.1.3 Sampling Interval

- Set the situation sampling interval equal to the collection interval of the data that is being monitored:
 - Example:
 - Cache collection done on 5-minute interval (default).
 - It is important to not set the sampling interval for a cache situation to 2 minutes.
 - Anytime you change the collection interval for an item, review and adjust the associated situations.
 - Do not set sampling interval for situations against the data set attributes database to less than once per day.

15.2 Data Collection

15.2.1 DASD Space and Fragmentation Collection

- Collect space data on only one image in a shared DASD environment.

- Use the exclude list, RKANPAR(U) member KDFDSCIN, to not monitor system volumes:
 - Page devices
 - JES spools
 - SYSRES volume(s)

15.2.2 VTS Collection

- Collect VTS data on a single image only

15.2.3 Dataset Performance Collection

Processor utilization associated with data-set level I/O monitoring shows up most prevalently in the IBM Subsystem address space. This is caused by the continual migration of data that is collected in the I/O exits to a shared data space accessible to the TEMS.

The first method of tuning this feature is to decide which volumes to monitor. You might not want or need to collect data-set level response time information for all of your volumes. Some of your volumes will probably require a better response time than others and it is possible to set different monitoring thresholds on each of your volumes.

There are two schemes available for collecting dataset level I/O statistics for a device:

The first scheme monitors I/O based on a fixed interval called sample count. This count specifies how often I/O events are monitored. If the sample count for a device is set to 1 (the default), every I/O is monitored. If the sample count is set to 5, every fifth I/O is monitored. Valid values are 1 - 99.

The second scheme activates device monitoring during periods of extended response time degradation and deactivates it during periods of acceptable response time. You specify an MSR trigger value for a device and a global trip count that controls the number of times an MSR exception occurs before monitoring resumes for the device. Valid values are 1 - 999.

Important: The values you employ in your test environment might lead to higher processor utilization when ported to your production image, even if you share DASD between the two. The migration of data collected in the I/O exits is directly proportional to the amount of I/O occurring on monitored volumes.

It is likely that your production environment incurs much more I/O than your test environment and if those I/Os are going to monitored volumes it will create more processing overhead. Of course, this is largely mitigated if you choose the second monitoring scheme, which only turns on monitoring when the trigger value is exceeded.

In summary:

- Do not use an asterisk (*) for volume mask and 1 to collect data for every I/O to every logical volume.
- Use exception monitoring to turn on collection when the volume response time is unacceptable.
- Monitor I/O for specific applications.

- Monitor every nth I/O to key volumes.
- Higher I/O rates on production systems will produce Higher processor use than might be seen on your test system even though your collection parameters are the same.

Executing the Storage agent on every system in a sysplex will result in a significant resource savings during DASD space and fragmentation collection. Space and fragmentation data will be collected only for those volumes that have changed since the last data collection.

RLS Dataset Groups

- Do not define the same RLS Dataset Groups for collection on multiple systems in a sysplex
- Do not collect RLS data for the same dataset for collection on multiple systems in a sysplex
- The data is available for the sysplex and for any system in the sysplex from any TEMS running in the sysplex (the storage agent need not be active on every z/OS image in the sysplex for this to be true)

15.2.4 Dataset Statistics Collection

- Data Set Groups:
 - Do not use masks in the high-level index.
 - Avoid collecting data for the same data set multiple times.
 - Do not collect space data for the same data set group on multiple systems.
- Data Set Attribute Database ad-hoc queries:
 - Include the volser in the predicate.
 - Include the data set name or mask.
 - Include the volser in situations against the Data Set Attribute Database.

16. OMEGAMON for Mainframe Networks

Collecting, processing and storing data consumes system and network resources. There are several variables that contribute to the amount of processing, memory and disk space used by IBM OMEGAMON for Mainframe Networks and the OMEGAMON platform components. The amount of processing, memory and disk space used on each monitored system depends on the number of resources being monitored, how often performance data is collected, and whether or not you choose to store historical data.

This chapter identifies options to consider while planning the configuration and deployment of this product to meet the needs of your enterprise. This chapter also provides information to consider when creating and modifying situations and workspaces.

16.1 Processor use for monitoring networks on z/OS systems

Total processor use attributed to running the IBM OMEGAMON for Mainframe Networks monitoring agent includes the percentage of processing used by the agent as well as the additional processor use associated with the following:

- Using the z/OS Communications Server Network Management Interfaces. As a result, you will notice a slight increase in the amount of processing used by the TCP/IP address space.
- Querying SNMP MIBs. This will be represented by processing incurred by OSNMPD. You must start an OSNMPD address space for each TCP/IP address space being monitored.
- Querying the IOASNMP or IOBSNMP subagent. The IOASNMP subagent OR the IOBSNMP subagent is required on systems where OSA performance data is being monitored.

Important: Remember that the IBM OMEGAMON for Mainframe Networks monitoring agent might not be the only application or user that is querying SNMP.

16.2 Understanding how real-time data is collected

Because of real-time data collection, you can monitor performance of your network and system resources in an effort to resolve potential problems before they affect the user.

The OMEGAMON for Mainframe Networks monitoring agent is designed to collect z/OS TCP/IP and VTAM® performance data. You must run the monitoring agent on each LPAR being monitored. The agent monitors all of the TCP/IP address spaces that are running on the LPAR.

The monitoring agent is capable of collecting several types of performance data, so you can monitor TCP/IP address spaces, TN3270 server sessions, High Performance Routing connections, Enterprise Extender connections, FTP sessions and transfers, OSA adapters, TCP/IP connections, interfaces, gateways, Communication Storage Manager, VTAM buffer pools, and VTAM environment. Performance data is collected using the z/OS Communications Server Network Management Interface (NMI) by querying SNMP MIBs and using the VTAM Performance Monitor Interface. The OMEGAMON for Mainframe Networks monitoring agent provides an efficient mechanism for collecting large amounts of network performance data.

Total storage associated with real-time data collection, as shown in Table 11.2, includes the storage used in the monitoring agent address space and the following data spaces used during data collection:

Table 11.2 Names and sizes for dataspace used during data collection

Dataspace name	Size	Description
KN3ACTCS	928 MB	This data space is used to store data that has been collected using SNMP, Performance Monitor Interface data, data collected using the z/OS Communications Server network management interface (NMI), the z/OS Communication Server external I/O control (IOCTL) function data, and Take Action command data. The storage area is reused each time the monitoring agent performs a collection. The actual amount of storage used is determined by the number of resources being monitored.
N3TCPIP	2 GB	This data space is used to store data collected using the z/OS Communications Server network management interface (NMI). A maximum of 225 MB is used for VTAM data. A maximum of 225 MB for each TCP/IP address space is used for TCP/IP data. The storage area is reused each time the monitoring agent performs a collection. The actual amount of storage used is determined by the number of resources that are being monitored.
N3FTP	2 GB	This data space is used to store data collected for FTP sessions and transfers. A maximum of 256 MB is used for each TCP/IP address space. Over time, all 256 MB is used because each new row of data is added to the existing data. When all 256 MB has been used, the data space wraps and new data is written to the beginning of the storage area.
N3TN3270	2 GB	This data space is used to store data collected for TN3270 server sessions. A maximum of 256 MB is used for each TCP/IP address space. The actual amount of storage used is determined by the number of resources being monitored. One record for each active session and one record for each session that closed in the last 24 hours is stored in the data space.

The data spaces are created when the monitoring agent initializes and are deleted when the monitoring agent is stopped. The sizes are not configurable. However, the amount of storage that is used is determined by the number of TCP/IP address spaces and the types of data that you configure the monitoring agent to collect.

16.3 Determining which systems and TCP/IP address spaces to monitor

All production LPARS and TCP/IP address spaces should be monitored because they represent the core of your business enterprise. You might choose to collect data less frequently on some systems, especially non-production LPARs, to minimize the cost of monitoring your networks.

16.4 Determining which types of real-time data to collect

By default, the OMEGAMON for Mainframe Networks monitoring agent is configured to monitor all resources (TCP/IP address spaces, TN3270 server sessions, High Performance Routing connections, Enterprise Extender connections, FTP sessions and transfers, OSA adapters, TCP/IP connections, interfaces, gateways, Communication Storage Manager, VTAM buffer pools, and VTAM environment). The monitoring agent always collects a required minimum amount of real-time data. You can disable one or more of the following optional types of data:

- TCP/IP Connection and Application Performance statistics collection
- Routing Table statistics collection
- TN3270 server statistics collection
- FTP data collection
- Enterprise Extender and High Performance Routing statistics collection
- Communications Storage Manager (CSM) buffer reporting
- Buffer Pool and VTAM Environment data collection

The following tables show the storage costs for monitoring the required and optional types of resources. These tables show you the relative size of attribute tables and the frequency that data is collected. You might use this information to determine what to monitor: which types of resources, which systems and at what collection interval.

The data shown in Table 11.4 is collected once every collection interval and stored in memory (in a data space). The memory is reused at each collection interval. When a user navigates to a workspace, a query results in the monitoring agent retrieving the appropriate data from a data space. Use this table to calculate the memory use by multiplying the row size by the number of resources. Calculate the total in the Memory Usage column to obtain the total memory used to hold data that is collected in an interval for a TCP/IP address space. Perform these calculations for each TCP/IP address space that you are monitoring.

Table 11.4 Data collected once every collection interval

LPAR Name				
TCP/IP Address Space				
Name Type of Data	Real-Time Data Attribute Table	Row Size in Bytes	Frequency Per Interval	Memory Usage
TCP/IP and VTAM (required collection)	TCPIP Address Space	569	1 row per TCPIP address space	
	TCPIP Devices	422	1 row per Device	

LPAR Name				
TCP/IP Address Space				
Name Type of Data	Real-Time Data Attribute Table	Row Size in Bytes	Frequency Per Interval	Memory Usage
	Interfaces	456	1 row per interface	
	OSA Express Channels	412	1 row per OSA channel	
	OSA Express LPARs	104	16 rows per OSA Channel per LPAR per local channel subsystem	
	OSA Express Ports	752	1 row per OSA channel per port	
	TCPIP Memory Statistics	392	1 row per TCP/IP address space	
	TCPIP Stack Layer	552	1 row per TCP/IP address space	
TCP/IP Connection and Application Performance statistics collection	TCPIP Applications	568	1 row per TCP/IP application	
	TCPIP Connections	536	1 row per TCP/IP connection	
	TCPIP Details	380	1 row per TCP/IP connection	
	TCP Listener	204	1 row per TCP listener	
	UDP Connections	304	1 row per UDP endpoint	
Routing Table Statistics Collection	TCPIP Gateways	404	1 row per TCP/IP gateway	

The data shown in Table 11.4.1 is collected once every collection interval and stored in memory. This data is collected for each LPAR that you monitor. The memory is reused at each collection interval. Use this table to calculate the memory use by multiplying the row size by the number of resources. Calculate the total in the Memory Usage column to obtain the total memory used to hold data that is collected in an interval for these resources.

Table 11.4.1 Data collected for each monitored LPAR

LPAR Name				
Type of Data	Real-Time Data Attribute Table	Row Size in Bytes	Frequency Per Interval	Memory Usage
TCP/IP and VTAM (required collection)	VTAM Summary Statistics	72	1 row	
Enterprise Extender (EE) and High Performance Routing (HPR) statistics collection	EE Connections	216	1 row per EE connection	
	EE Connections Details	212	5 rows per EE connection	
	HPR RTP Connections	536	1 row per HPR RTP connection	
Communications Storage Manager (CSM) buffer reporting	CSM Storage	112	1 row	
Buffer Pool and VTAM Environment data collection	VTAM Address Space	244	1 row	
	VTAM I/O	72	6 rows	
	VTAM Buffer Pools	156	14 rows	
	VTAM Buffer Pool Extents	96	1 row per buffer pool extent	
	VTAM Buffer Usage by Address Space	72	2 rows per address space using IO00 or CRPL buffers	
	VTAM Buffer Usage by Application	80	1 row per application per address space using IO00 buffers	
	VTAM Buffer Usage by Category	68	12 rows	

The FTP data shown in Table 11.4.2 is collected when a new session or transfer is opened or when an existing session or transfer is closed. This data is collected when z/OS Communications Server notifies the monitoring

agent when data is available and therefore does not adhere to a collection interval. As explained previously, new records are appended to the already collected data until the table in the data space is full, at which time the table wraps. Therefore, over time 256 MB per TCP/IP address space will be used to hold FTP data.

This data is collected for each TCP/IP stack where FTP is running.

Table 11.4.2 FTP data collected

LPAR Name				
TCP/IP Address Space Name				
Type of Data	Real-Time Data Attribute Table	Row Size in Bytes	Frequency	Maximum Rows Stored
FTP Data Collection	FTP Sessions	348	2 rows per FTP session	25 000
	TCPIP FTP	2420	2 rows per FTP transfer	100 000

The TN3270 session workspaces display information about open, closed and active TN3270 sessions for a TCP/IP address space. Data for open and closed sessions is provided when z/OS Communications Server notifies the monitoring agent that data is available and therefore is not driven by a collection interval. On an LPAR running z/OS 1.8 or higher, data for active sessions is collected once per collection interval.

Memory used to store data for one session is reused for the same session at each collection interval and for the data collected when the session is closed. Approximately 24 hours after a session is closed, the memory that is used to hold the data for that session is made available for a new session.

Use Table 11.4.3 to calculate the memory use by multiplying the row size by the number of resources. Total the Memory Usage column to obtain the total memory used to hold TN3270 data collected for a TCP/IP address space. Perform these calculations for each TCP/IP address space that you are monitoring.

Note that the TN3270 Response Time Buckets table is not collected or stored as a separate table. Instead, it is a different view into the TN3270 Server Sess Avail table. When a query is issued to retrieve TN3270 Response Time Buckets data, the appropriate TN3270 Response Time Buckets rows (one row for each of the five response time buckets) are created from the corresponding row in the TN3270 Server Sess Avail table.

[*Table 11.4.3 TN3270 data collected*](#)

LPAR Name				
TCP/IP Address Space Name				
Type of Data	Real-Time Data Attribute Table	Row Size in Bytes	Frequency	Maximum Rows Stored
TN3270 Server Statistics Collection	TN3270 Server Sess Avail	400	1 row per TN3270 server session that is active or was closed in the last 24 hours	
	TN3270 Response Time Buckets	204	0 rows	0

Use the Configuration Tool or the z/OS MODIFY command to enable or disable data collection for specific types of data. Refer to the KN3FCCMD commands appendix in the *OMEGAMON for Mainframe Networks: Configuration Guide* book.

OMEGAMON II for Mainframe Networks is capable of collecting TCP/IP performance data. However, this option is disabled by default. OMEGAMON for Mainframe Networks provides more extensive coverage of TCP/IP and its resources and provides a more efficient monitoring solution. Be aware that enabling both OMEGAMON II for Mainframe Networks and OMEGAMON for Mainframe Networks to monitor TCP/IP generates unnecessary processing overhead. Use OMEGAMON for Mainframe Networks to monitor TCP/IP resources in your enterprise.

16.5 Defining data collection intervals

The collection intervals control how often real-time data is collected. Each time that you collect data, the monitoring agent incurs a certain amount of processor use. How much processing is incurred depends on the number and type of resources you monitor.

Collection intervals are specified in minutes and have a valid range of 1 - 60. The default value for a collection interval is 5 minutes. This default satisfies the monitoring needs of many customers who require near “real time” access to data.

The collection intervals are specified in the configuration tool or you can modify a collection interval by issuing a MODIFY *procname*,KN3FCCMD command (refer to the *OMEGAMON for Mainframe Networks: Configuration Guide* for more information about the MODIFY commands). You specify values for the following collection intervals:

- TCP/IP sample interval: the collection interval used to control the majority of the data collection, with the exception(s) that follow.

- SNA data collection interval: controls the Buffer Pool and VTAM Environment data collection. This option is available only if you enter Y for Buffer Pool/VTAM Environment Data Collection when you configure the monitoring agent in the Configuration Tool.

The z/OS Communications Server Network Management Interface and Performance Monitoring Interfaces maintain performance information between collection intervals. For example, spikes are identified in such attributes as "Maximum_Authorized Private_Storage Allocated". Over- and under-utilization is identified in such attributes as "Total Bytes Sent."

The network performance data that is displayed in the TEP is the last collected; a user's request to view data does not drive a new data collection. Therefore, increasing the frequency of data collection (that is, decreasing the collection interval) by the monitoring agent generally results in your network operators viewing more recent data but results in higher processor consumption by the monitoring agent. By decreasing the frequency of data collection (for example, from 5 to 15 minutes), you reduce the number of times per hour that the monitoring agent collects data (from 12 to 4 times per hour), thus reducing the total processor consumption (by a factor of approximately 3).

You can further reduce processor consumption by changing the Routing Table Collection Frequency in the Configuration Tool or on a KN3FCCMD START ROUTE command. Refer to the KN3FCCMD commands appendix in the *OMEGAMON for Mainframe Networks: Configuration Guide* book for information about this command. The Routing Table Collection Frequency controls how often the Routing Table Statistics (Gateway table) are collected. By default, this data is collected once every 10 collection intervals (once every 50 minutes when the default collection interval is used). The frequency does not affect the amount of storage used; it affects only processor consumption.

Set the collection intervals and routing table frequency to meet your monitoring needs, ensuring that you are promptly alerted to problems so that they can be resolved quickly. But, be aware that each data collection comes with a cost in processor use.

Run OMEGAMON for Mainframe Networks and use OMEGAMON on z/OS or RMF to measure processor use. Adjust the collection intervals until you achieve the preferred balance between currency of data and processor use.

16.6 Defining display intervals

During configuration of the monitoring agent, you specify values for FTP and TN3270 display intervals. A display interval determines how long a session or transfer is available for display at the TEP. The value is specified as a number of hours (1 - 24 are valid values). FTP and TN3270 workspaces display session and transfer information that was collected during the specified display interval. Note that some workspaces also have default query filters that determine which data or how much data to display.

Setting a display interval to a lower value might reduce the amount of processor use by the monitoring agent when retrieving data for display at the TEP. Similarly, setting a display interval to a higher value might increase the amount of processor use. Balance your need to view data for sessions or transfers that occurred up to 24 hours ago with the processor use.

An alternative to increasing the display interval is to enable historical collection for the data table and by viewing historical data instead of real-time data in the TEP. Refer to the *OMEGAMON for Mainframe Networks: User's Guide* for more information about the display intervals used for FTP and TN3270 workspaces.

16.7 Defining and running situations

Situations are used to identify monitored resources that meet certain performance criteria, raising an alert when the criteria are met. A situation definition includes a sampling frequency, a set of conditions, and a list of monitored systems. Each of these has implications on processor and storage consumption. Also, the cumulative effect of all the active situations has implications on performance.

OMEGAMON for Mainframe Networks provides 83 predefined situations (17 that are automatically started) to detect some of the most common mainframe network problems. Automatically started situations run automatically when the OM MfN monitoring agent is started. The automatically started situations provided with the product represent the situations that generate critical alerts. Refer to *OMEGAMON for Mainframe Networks: User's Guide* for a list and description of the product-provided situations.

When planning for configuration and deployment of OMEGAMON for Mainframe Networks, evaluate all situations provided by the product. Determine which situations to start automatically and eliminate situations that are not relevant to your monitoring strategy. If necessary, modify existing situations and create new situations to meet the needs of your enterprise.

For each situation that you choose to run, determine the importance and therefore the preferred sampling frequency. Selecting several different intervals can increase the cost without increasing the benefit. Consider the attribute table that a situation queries. A good way to optimize the performance of situations is to enable the OMEGAMON platform to group situations.

For situations to be grouped, they must be active when the hub TEMS starts, have the same sampling frequency, and test conditions on attributes in the same attribute table. You want to start automatically all high-severity situations on an attribute group with the same sampling frequency. You might also be able to gain efficiencies by automatically starting all situations on an attribute group with the same sampling frequency, regardless of severity.

Verify that the conditions that are evaluated by each situation are appropriate for your environment. Check both the set of conditions and individual conditions. The predefined situations attempt to use the most efficient means to identify problems. There might be alternative conditions that identify the same problems but are less expensive to evaluate in your environment. Ensure that the values being checked are correct for your environment.

After you modify situations that are automatically started, stop and start the hub TEMS. The process of combining situations occurs only during initialization of the hub TEMS.

16.8 How historical data is collected

Historical data collection is an optional function that is enabled using the TEP. The OMEGAMON platform provides the following types of historical data collection:

- Short-term historical data is stored in the persistent data store on z/OS systems or in files on distributed systems. To optimize performance, configure the persistent data store at the monitoring agent, meaning that you will have a persistent data store on each z/OS system that you are monitoring.

Short-term historical data typically refers to data that is stored for 24 hours or less. However, the amount and age of the short-term data that is retained depends on the number of resources being monitored and the amount of disk space configured for use by the persistent data store.

- Long-term historical data is stored in the TDW, which is located on a Windows® computer. The long-term history database can retain data collected by OMEGAMON for Mainframe Networks monitoring agents for as long as you like (days, weeks, months or years).

Short-term historical data is best used for analysis during problem determination. Additional prerequisite software is not required for short-term historical data collection, however the data sets that are used by the persistent data store must be configured using the Configuration Tool.

Long-term historical data is better used for trend analysis and to determine workload balance. While the supported database product for TDW with OMEGAMON Version 360 was Microsoft® SQL Server, IBM Monitoring also supports IBM DB2 Universal Database™ and Oracle databases. Refer to the chapter about configuring the warehouse proxy for TDW in the *IBM Monitoring: Installation and Setup Guide* for the list of supported databases, releases and operating system platforms. Long-term history also requires installation of the warehouse proxy software (provided) and configuration of an Open Database Connectivity (ODBC) connection. Use the warehouse proxy installation default support for defining database tablespaces and creating the ODB connection. Short-term historical data collection must be enabled and configured if you want to perform long-term historical data collection.

After historical data collection is enabled, an icon is displayed in qualifying views in TEPworkspaces. Allow time for historical data to be stored, to produce meaningful reports. You can click this icon to extend any existing TEPview (also called a report) to include historical data. TEPreports automatically pull data from short-term and long-term history, based upon the time period you specify for the report.

The collection interval for historical data can be configured to be different than the collection interval for real-time data. To avoid processing overhead and to decrease storage consumption, historical data collection is typically performed less frequently than real-time data collection. You can configure a short-term historical data collection interval of 5, 15, 30, or 60 minutes.

Writing the data to long-term history can be configured for 24 hours, 1 hour, or OFF. If you configure long-term history, we recommend a warehousing interval of 1 hour to avoid transferring 24 hours worth of historical data at one time. This shorter interval reduces the duration of processor use associated with writing data to the warehouse by spreading the writing across 24 periods.

The following example illustrates the rate of accumulation of historical records, assuming the following intervals are specified:

- Real-time data collection interval: 5 minutes
- Short-term historical data collection interval: 15 minutes
- Long-term warehousing interval: 1 hour

- At 1:57, data collection is initiated for 1 TCP/IP Address Space.
- One row of data (512 bytes) is collected at 1:57, another row at 2:02, another at 2:07, 2:12, 2:17, and so on. Only the current row of data is stored because the old row is deleted when a new row is created. Because the real-time data collection interval is 5 minutes, 12 collections are made per hour.
- At 1:58, collection of historical data is initiated for 1 TCP/IP address space.
- One row of data (540 bytes) is stored in short-term history at 2:00, the second row is collected at 2:15, the third at 2:30, 2:45, 3:00, and so on. Short-term history uses the most recent collection and therefore does not initiate another data collection. The row stored at 2:00 would use the 1:57 collection. The row stored at 2:15 would use the 2:12 collection.
- Because the short-term historical data collection interval is 15 minutes, four collections are made per hour. All measurements are stored for future use.
- After one hour, all (four) rows of short-term historical data are transferred by the warehouse proxy to the long-term history SQL database.
- After 24 hours, 96 (24 x 4) rows of data is stored in the TDW.

16.9 Determining which types of historical data to collect

When deciding which types of data to store in short-term and long-term history and how long to store it, you need to recognize that data collection consumes processor cycles and disk space. Writing data to short-term history is cost effective and typically much less costly than writing to long-term history. Retrieving short-term history data to display historical reports increases processor use at the hub TEMS and at the monitoring agent, when short-term history data is stored at the monitoring agent.

Short-term historical data is written to disk, typically performed at the monitoring agent, consuming processor cycles on the z/OS monitoring agent system. Additional processor cycles are used when the Warehouse Proxy extracts data from short-term history and transfers it to the Data Warehouse. When you collect a large amount of data in short-term history, the extraction process significantly increases the processor use of the monitoring agent. Similarly, when a table contains thousands of rows, the retrieval process significantly increases the processor use of the monitoring agent.

For the best utilization of processor and storage resources, do the following actions:

- Short-term history location: persistent data store defined at the z/OS agent.
- Long-term history location: a dedicated Windows server. If you are monitoring fewer than 5 000 resources, you might prefer to use the same Windows system as the TEPS.
- Short-term history collection interval: 15 minutes or less frequently.
- Warehousing interval: hourly.

Depending on your needs, you may configure historical data collection for only a subset of attribute tables. This is an effective means for limiting storage and processor consumption, particularly if you choose not to perform historical data collection for high volume attribute tables such as TCP connections or attribute tables with many bytes per row (many attributes), such as FTP transfers. Collect only data that you will use in historical reports. Collect those tables only as frequently as your enterprise needs. Selecting a less frequent historical collection interval (30 minutes instead of 15 minutes) will reduce both storage and processor consumption.

Using the tables in the [Determining which types of real-time data to collect](#) section you can calculate storage consumption based on real-time data collection. The information you gather for these tables also provide the basis for calculating the storage requirements for historical collection.

The additional storage cost for short-term historical data collection equals the number of bytes of real-time data storage, per row of data, plus 28 bytes. The additional storage cost for long-term historical data collection, in the TDW, is the same number of bytes. To calculate storage associated with historical data, add 28 bytes to each row of real-time data. Refer to the historical collection appendix in *Configuring IBM OMEGAMON for Mainframe Networks* for detailed calculations.

You can use this information as a basis for choosing which attribute tables to enable for historical collection. You can select individual attribute tables for historical collection, including specifying different historical collection intervals and warehouse intervals. Here is a list of attribute tables that are likely to have a large number of resources and therefore require higher amounts of storage and processor consumption when enabled for historical collection

- TCPIP Connections
- TCPIP Details
- TN3270 Server Sess Avail
- TCPIP Devices
- Interfaces
- EE Connections
- EE Connections Details
- HPR Connections
- VTAM Buffer Pool Extents

If you choose to not collect all attribute tables, TCPIP Connections is a logical choice to omit. All of the data in TCPIP Connections is also available in TCPIP Details, TCP Listener, and UDP Connections. This choice affects which workspaces can draw historical reports, but the data is there if needed for reporting.

By default, historical reports retrieve up to 24 hours of data from short-term history. If your persistent data store is not allocated with sufficient space, you will not have 24 hours of short-term data to retrieve. Allocate your persistent data store to hold a full 24 hours of data or change the default of 24 hours. You might also want to change the default in order to retrieve less data from short-term history and more data from long-term history in order to reduce the processing consumed by the monitoring agent to process queries for historical data. Refer to *Configuring IBM OMEGAMON for Mainframe Networks* for information about how to change the default of 24 hours.

Because historical data accumulates, you must also determine how long you want to keep the data. Short-term history data in the persistent data store automatically wraps, and thus does not need to be maintained. You can also run a KPDXTTRA job to write short-term history to flat files, for backup, or for analysis in a statistical or graphing package. Refer to *IBM Monitoring: Administrator's Guide* for details on about the KPDXTTRA job.

Long-term history, in a SQL database, does not automatically prune old records. You need to determine how much data to retain, and either use the database manager tools to manually delete old records or schedule a script to run automatically to delete old records.

For detailed instructions about setting up historical data collection, refer to the section on configuring your system for TEP in *IBM Monitoring: Installation and Setup Guide*. Additional agent configuration information about the warehouse proxy is found in *IBM Monitoring: Administrator's Guide*. For information about reporting, refer to *IBM Monitoring: User's Guide*.

16.10 Designing workspaces

When a user navigates to a workspace, one or more queries are processed by the OMEGAMON components in order to display the requested workspace. Those same queries are processed again when the user requests a refresh or periodically in the cases where the workspace is configured to refresh automatically.

The workspaces and queries provided in the OMEGAMON for Mainframe Networks product have been designed with performance in mind. However, your environment and the resources that you monitor might require customization of the product-provided workspaces and queries.

The following tips can improve the performance for users viewing workspaces. Refer to *Using IBM OMEGAMON for Mainframe Networks* to see the default filters that are provided for each workspace.

The query assigned to a chart or table view requests data from a particular attribute table. It runs when you open or refresh the workspace. The TEPs send the query to the hub TEMS. The hub TEMS distributes the query to the appropriate monitoring agent or agents and aggregates the resulting rows. The TEPs retrieve the results and holds the entire result set in memory. The TEP retrieves one page of the results to display and holds both the current and previous page in memory.

You can dramatically reduce the amount of data retrieved by doing the following things:

- Reducing the number of rows or attributes retrieved
- Applying the same query to multiple views in a workspace
- Adjusting the auto-refresh rate

16.10.1 Reducing the number of rows retrieved

One of the best ways to improve the performance of queries is to reduce the number of rows retrieved. You can use the Query Editor to add filters that reduce the number of rows that are returned. You might want to change the existing filter values of a query or add filters to the query. For example, the Applications workspace contains a table view that displays all applications that have TCP/IP connections. You might be interested in only those applications that have active connections. You can customize the query by adding a filter in the Query Editor for "Active Connections" greater than 0. Refer to *Using IBM OMEGAMON for Mainframe Networks* to see the default filters provided for each workspace.

Do not confuse custom queries with view filters, which can also be started from the TEP properties window. View filters have no effect on reducing processor and storage consumption by the monitoring agent and actually increase the TEPclient processor consumption.

View filters are applied by the client and affect only the current page. If more than one page is returned by the query, only a subset of the data is viewed on each page. Increasing the page size is an option available in TEP. This typically provides more filtered data on each page, but increases the client's memory requirements because now the two pages per query stored at the client are larger. It is more efficient to filter in the queries

16.10.2 Reducing the number of attributes retrieved

Most product-provided queries return all attributes. There might be 50 attributes in an attribute table, yet you might want to view only 25 of them. Creating a custom query to retrieve only those 25 attributes reduces TEPS and client processing and memory requirements.

For example, the OMEGAMON for Mainframe Networks applications attribute table contains 54 attributes. If you are creating a workspace that displays information about TCP applications, there is no need to select the 14 UDP attributes of the attribute table.

Historical workspaces retrieve much more data than real-time workspaces. Accordingly, all of the queries that are used for predefined historical workspaces request only the most valuable subset of attributes on the source table. The queries that are used by predefined historical workspaces are good starting places for building better performing workspace views. Refer to *Using IBM OMEGAMON for Mainframe Networks* for more information about modifying or creating queries.

16.10.3 Applying the same query to multiple views in a workspace

Having multiple views in a workspace that retrieve data from different attribute tables is fine. But if you have a graph containing "Total Retransmissions" and a table containing "Transmit Segment Rate" (both available from the same attribute table), create one custom query for both views. By creating a single custom query, TEP retrieves the data once for both views.

The objective is to use only one query for each attribute table used in a workspace. When a workspace is displayed, the entire results set for each query is stored on the TEPS. The 100 rows (default page size) from each query currently being viewed and the previous page of any pane viewed are stored on the TEPclient.

16.10.4 Adjusting the auto-refresh rate

The operator can choose an automatic refresh rate from every 30 seconds to once per hour. Each time the workspace is refreshed, the data is retrieved from the data spaces that reside on the system where the monitoring agent is running. This happens only for the currently displayed workspace. Retrieving data from the agent consumes processing power, so it is important to specify a refresh rate that meets your monitoring needs while avoiding unnecessary performance overhead by the monitoring agent.

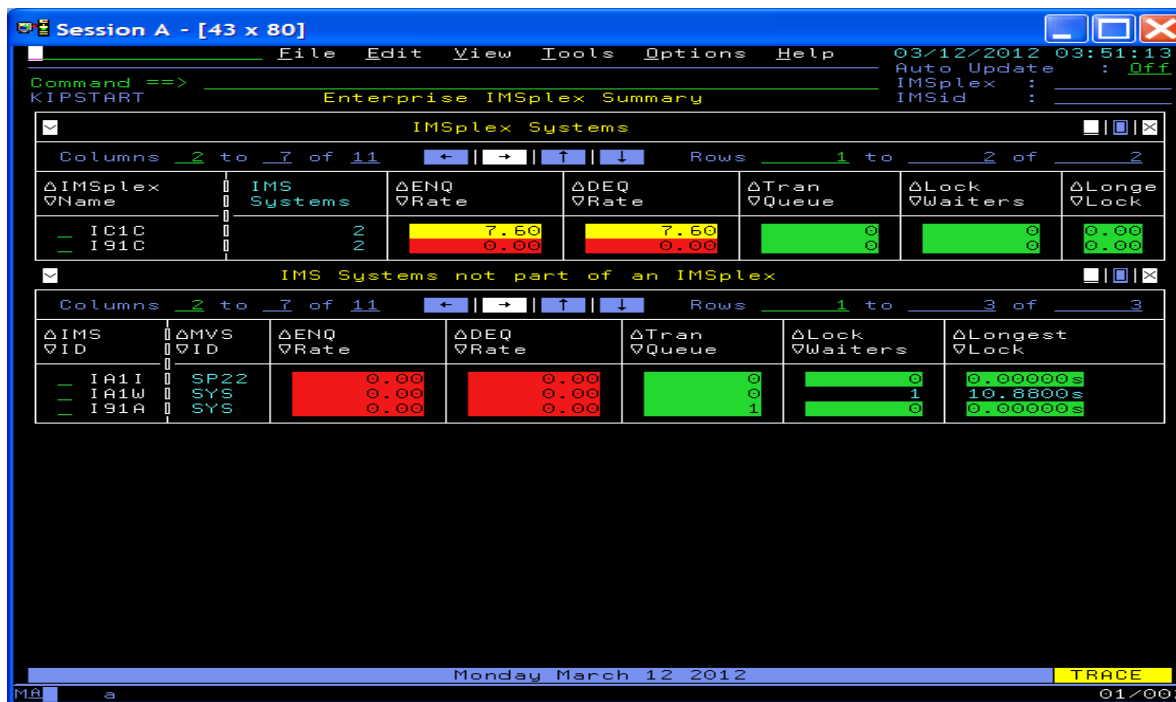
When an operator clicks the Time Span button to display historical data, the auto-refresh rate defined for the workspace continues to be used. This might result in significant processing consumption by the monitoring agent as the workspace is automatically refreshed (as frequently as once every 30 seconds). Consider changing the default auto-refresh rate to "On Demand" for workspaces that your users frequently use to display historical data.

17. OMEGAMON 3270UI Product Guidelines

17.1 OMEGAMON 3270UI

The new, comprehensive 3270 user interface (3270UI) allows integration of enterprise-wide views on a single screen along with simplified reporting.

- Support Teams need training for 3270UI for usage and problem determination.
- 3270UI provides faster problem diagnosis by requiring fewer steps to find root cause: 50-75% of time needed to solve problems.



17.2 Enhanced 3270 Interface

With all components, active, the Enhanced 3270 interface should be used by logging on to VTAM APPLID defined for the 3270UI address space. Authentication is done via the security product on the system the 3270UI address space is running on. Authorization for specific actions is not active by default, but can be implemented via defining security classes in the security product and adding security class information in the RKANPARU(KOBENV) member. The data displayed comes from the Hub TEMS defined for the user, based on either a site wide default or a specific definition for a particular user. The initial screen is the default workspace overview of the current Sysplexes and CICSplexes the DRAs have discovered from the Hub TEMS:

IBM z Systems IBM Monitoring – OMEGAMON Best Practices & Performance Guidelines V2

Enterprise Summary

05/14/2012 11:17:20
Auto Update : Off
Plex ID :
Sys ID :

All Active Sysplexes

ΔSysplex Name	ΔAverage CPU Percent	Highest LPAR Name	ΔHighest LPAR CPU%	ΔPercent LPAR VMSU Capacity	+LPAR Group Name
WSCZPLEX	1	ROSP2	1	0.4	N/A
HIAVSYSL	3		4	1.6	N/A

All Active CICSplexes

ΔCICSplex Name	ΔNumber of Regions	ΔTransaction Rate	ΔCPU Utilization	Any SOS Regions	SOS Region
OMEGPLEX	2	6/m	0.0%	No	n/a

CICSplex groups are initially created based on CICSplex definitions in CICSplex Systems Manager (CPSM). If CPSM is not being used, or a CICS region is not defined to CPSM, it is placed by default into the OMEGPLEX group. The rules of creating custom groups and assigning CICS regions to them can be changed via this interface

From this point the screen panels can be scrolled to see more data elements (the arrows highlighted in white indicate when scrolling in that direction will display additional columns or rows). The Sysplex and CICSplex names can be selected for further drill down. In this example, the top resource consumers across a sysplex can be seen in one view:

Top Consumers for Sysplex WSCZPLEX

05/14/2012 12:12:03
Auto Update : Off
Plex ID :
SMF ID :

Highest Consuming Address Spaces of Real Storage

ΔAddress Space Name	ΔASID	ΔCentral Frame Count	Working Set Size	ΔLPAR Name
XE4CDSST	004B	125039	500156K	ROSP2
ZFS	0033	50933	203732K	ROSP2
RMF	0024	38065	152260K	ROSP2

Highest Consuming Address Spaces of Virtual Storage

ΔAddress Space Name	ΔASID	ΔTotal Virtual(Mb)	ΔTotal Fixed(Mb)	ΔLPAR Name
GRS	0007	241967.0	1.2	ROSP2
DSNCDBM1	0057	9451.0	4.6	ROSP2
HZSPROC	002C	8220.2	0.6	ROSP2

Highest Consumers of CSA Storage

ΔAddress Space Name	ΔASID	ΔCSA In Use	Δ% of Total CSA	CSA Orphaned	ΔLPAR Name
SYSTEM	0000	122880	3.6	No	ROSP2
MASTER	0001	78848	2.3	No	ROSP2
MXIMAST	002B	45056	1.3	No	ROSP2

The next two pictures are examples of a CICS region summary and a transaction group service level summary within the CICSplex group:

IBM z Systems IBM Monitoring – OMEGAMON Best Practices & Performance Guidelines V2

```
File Edit View Tools Options Help 05/14/2012 14:32:25
Auto Update : Off
Command ==> CICSplex Regions Summary CICSplex : OMEGPLEX
Region :
KCPRGNS
```

Regions Summary for OMEGPLEX					
ΔCICS Region ▽Name	ΔCPU ▽Utilization	ΔTransaction ▽Rate	ΔMaximum Tasks ▽Percent	SOS	ΔStg. Violat ▽
— CICS31A	0.0%	12/m	20%	No	0
— CICS31B	0.0%	6/m	20%	No	0

```
File Edit View Tools Options Help 05/14/2012 14:33:59
Auto Update : Off
Command ==> CICSplex Service Level Summary CICSplex : OMEGPLEX
Region :
```

CICSplex Service Level Analysis for OMEGPLEX				
Service Class Name	Transaction Rate	Interval End Timestamp	% Time Using CPU	% Wait on DB2
— CTRANS	0	14:30:00	7%	0%
— LTRANS	5	14:30:00	45%	41%

Drilling down to particular resources provides the capability to take actions against them (security can be applied to restrict this function, and this is strongly recommended). Here is an example of the actions that can be taken against a z/OS address space:

```
File Edit View Tools Options Help 05/14/2012 19:25:51
Auto Update : Off
Command ==> KM5ASPO KM5ACT Take Actions for Address Space CSQ3MSTR
: HIAVSYS
: HCB$
```

Take Actions for Address Space CSQ3MSTR						
Select an action and then press ENTER						
Address	Started T	TSO User	Batch Job	Columns	ΔAddress S ▽Name	
— R41AMC					0055	2.2
— R41ATOM					0057	2.2
— CSQ3MSTR					005C	1.3
— CSQMMSTR					0006	0.4
— R41ADSST						
— XCFAS						

Customization

Customization can be done to the following areas:

1. Interface appearance: profiles can be created to customize the appearance of the workspace panels. These interface profiles allow for customization, down to the user level, of the following attributes:

- Locale (e.g. date/time format, currency format, separators, etc.)
- Headers, borders, and text colors
- Status indicator severity
- The initial screen displayed after a user signs on
- The target Hub TEMS from which data is collected

Customized profiles are defined as members in the UKOBDATF dataset. A default profile is provided in the TKOBDATF dataset, and can be used to create a site profile (a member named CUASITE) that with site-wide settings different from the default profile. User profiles (members named after the user ID) for individual user settings can also be created.

Customized profiles are defined as members in the UKOBDATF dataset. A default profile is provided in the TKOBDATF dataset, and can be used to create a site profile (a member named CUASITE) that with site-wide settings different from the default profile. User profiles (members named after the user ID) for individual user settings can also be created.

2. Workspace configuration: workspaces can be created to display specific information to meet local monitoring requirements. For example, a workspace can be created to display data from different OMEGAMON products. Items that can be customized for a workspace include

- The layout of the workspace panel and contained subpanels
- Data queries to retrieve data for display in the workspace
- Data column display attributes
- Data filters
- Amount of data rows to display

Customized workspace definitions are defined as members in the RTE UKANWENU dataset.

3. Threshold settings: screen information can be highlighted based on the values. Each product provides a default set of thresholds as a member (KppTHRSH, where pp = the 2 character product code) in the TKANPAR dataset. These can be copied into the RTE RKANPARU dataset, modified as needed and refreshed dynamically. Thresholds can be applied site wide (member name CUASITE) or for individual user views (member named after the user ID).

The threshold specification follows an IF... THEN... syntax, where the IF clause can evaluate the values, and the THEN clause sets a status state (GOOD/WARNING/CRITICAL/IDLE/NOSTATE) and a status value (numeric 0-9 or NORANGE). For example, the CICS CPU Utilization in the CICS Regions Overview workspace above is displayed in red due to this threshold definition

```
IF ( OMCICS.CICSROV.CPUUTIL GT 90.0% OR
    OMCICS.CICSROV.CPUUTIL LT 5.0%
)
THEN DO
STATUS ( CRITICAL 9 )
ENDDO
```

The display value is red since the default interface profile maps red as the color for a

CRITICAL status.

Details of implementing these customization options can be found in the *IBM Monitoring OMEGAMON Enhanced 3270 User Interface Guide*.

18. OMEGAMON Presentation Best Practices

18.1 TEPs Presentation Best Practices

Several overall presentation guidelines should be utilized, regardless of the OMEGAMON products being used or the environment being monitored. Create simplified custom navigators, rather than using the physical navigator for every monitoring audience. A variety of custom navigators can be developed for different audiences. These custom navigators should be used to make it very easy for each audience to get the information they want to see quickly and with a minimum of navigation.

OMEGAMON on z/OS can provide overview information regarding CPU utilization, storage utilization and any outstanding ENQ or Reserve problems. Other OMEGAMON monitoring agents can provide high-level information about the CICS, and DB2 subsystems. A Situation Event Console view should be included to provide information about all open events.

Child navigator items, one level below the top navigator item, provide more detailed information about the status of each of the subsystem types. This is merely one example of the type of navigators and overview workspaces that can be built for simplified monitoring. The key to building effective custom navigators and overview workspaces is to first determine the requirements of the monitoring audience and then understand how to use the OMEGAMON software facilities to fulfill those requirements.

Realistic situations should be built and distributed in the OMEGAMON product to monitor the customer's environment. New features in the current version can be used as opposed to using the older legacy classic product to monitor the production environment. The OMEGAMON product can be used to monitor applications across platforms or across products to give a view of how the business is being affected by a system or application problem or issue.

The alerts need to be based on accurate and realistic situations. Business and or application based situations and views need to be created. This is the process on combining attributes across OMEGAMON products anywhere the application runs that monitors are in place. This would create an end-to-end approach to monitoring. More complex situations and views would create a business impact view to see how the business is being impacted and where.

Building situations with persistent settings will reduce alerts based on very short spikes opposed to true longer problems that affect business.

Best practices for the use of OMEGAMON products can be described in three categories – Presentation, Alerting and Tuning.

IBM z Systems IBM Monitoring – OMEGAMON Best Practices & Performance Guidelines V2

Presentation includes the creation of custom navigators, workspaces, views and queries that allow OMEGAMON monitoring information to be presented in a way that will provide an efficient and effective display of enterprise monitoring data.

Alerting includes the creation of custom situations, and sometimes policies that will provide notification of problems in the enterprise that need to be investigated. Optionally, problem alerts may be sent to an enterprise event console, and automated actions may be initiated by situation events.

Tuning is a methodology as much as it is a “practice”. A simple methodology could be represented as follows:

1. Begin collecting historical data for all pertinent systems/LPARs. This can be RMF reports, SMF data, and/or OMEGAMON performance data stored in the TDW.
2. Set thresholds based on current service level agreements. If no service level agreements exist, then use historical data to help construct realistic thresholds. The TCR application and IBM Performance Analyzer can help produce the required reports.
3. Set thresholds according to the results of #2.
4. When alerts are triggered determine if the alerts are valid. If they are, respond to the alert appropriately. If the alerts are not valid, reset the situation threshold to a more appropriate value.
5. Go back to #4

Several overall presentation guidelines should be utilized, regardless of the OMEGAMON product being used or the environment being monitored. Each of these techniques will be discussed in greater depth in the following sections.

- Create simplified custom navigators, rather than using the physical navigator for every monitoring audience. A variety of custom navigators can be developed for different audiences. These custom navigators can be used to make it very easy for each audience to get the information they want to see quickly and with a minimum of navigation.

The following screen shows an example of one possible custom navigator and a high-level overview workspace designed for an Operations group.

The navigator is organized by major subsystems that Operations is responsible for monitoring. OMEGAMON on z/OS is providing overview information regarding CPU utilization, storage utilization and any outstanding ENQ or Reserve problems. Other OMEGAMON monitoring agents are providing high-level information about the CICS, DB2, IMS and MQ subsystems. A Situation Event Console view is included to provide information about all open events.

Child navigator items, one level below the top navigator item, provide more detailed information about the status of each of the subsystem types. Additional examples will be shown in later topics.

This is merely one example of the type of navigators and overview workspaces that can be built for simplified monitoring. The key to building effective custom navigators and overview workspaces is to first determine the requirements of the monitoring audience and then understand how to use the OMEGAMON software facilities to fulfill those requirements. More information about building custom navigators and workspaces can be found in the [IBM Monitoring User's Guide](#). A current version of this manual is located here: http://publib.boulder.ibm.com/infocenter/tivihelp/v15r1/topic/com.ibm.itm.doc_6.2.1/itm_user.htm .

OMEGAMON on z/OS will be used for the Presentation Best Practices sample workspaces.

The screenshot displays the Tivoli Enterprise Portal interface with several monitoring panels:

- Top 10 CPU Users:** A table listing jobs and their CPU usage.

Job Name	Step Name	JESJOBID	CPU Percent	Job Elapsed Time	TCB Percent
DNET687	DBAUSER	TSU04660	33.4	2,353.29	33.4
BBOS004S	BBOS004S	STC17958	22.1	1,208,935.84	0.4
CACAF00	CACAF00	STC16226	3.0	1,848,708.92	3.0
CXEG02	CXEG02	STC04110	3.0	93,139.39	3.0
DBA031	DBAUSER	TSU04669	2.6	150.62	2.6
NETVIEW	NETVIEW	STC16866	1.3	5,496,420.50	1.3
CXEGH2	CXEGH2	STC16878	1.3	5,496,420.50	1.3
CXEGH2RC	CXEGH2RC	STC16879	0.8	5,496,420.50	0.8
RMPGAT	RMPGAT	STC14895	0.4	7,510,736.51	0.4
- Common Storage Utilization:** A table showing storage area usage.

Managed System	Area	Allocation	Allocation Percent	In Use	Ir Pr
DEMOPLX.MVSB.MVSSYS	CSA	380928	10	303104	
DEMOPLX.MVSB.MVSSYS	ECSA	44511232	12	44080128	
DEMOPLX.MVSB.MVSSYS	SGA	0	0	506880	
DEMOPLX.MVSB.MVSSYS	ESGA	0	0	18024576	
DEMOPLX.MVSC.MVSSYS	CSA	360448	10	287744	
DEMOPLX.MVSC.MVSSYS	ECSA	35414016	10	35035136	
DEMOPLX.MVSC.MVSSYS	SGA	0	0	503908	
DEMOPLX.MVSC.MVSSYS	ESGA	0	0	15126528	
DEMOPLX.MVSA.MVSSYS	CSA	2404352	64	2182144	
- Situation Event Console:** A list of system events with severity levels (Warning, Critical) and status (Open).
- Enqueue and Reserve Data:** A table showing address space details for various system names.
- DB2 Lock Conflicts:** A table detailing lock status, elapsed time, and resource usage for DB2 identifiers.
- IMS System Exceptions:** A table listing system identifiers and timestamps for IMS-related exceptions.
- CICS Region Overview:** A table providing an overview of CICS regions, including system ID, region name, and version.
- DB2 System Status:** A table showing the status of DB2 systems, including thread counts and tape mount status.
- MQ Channel Performance:** A table detailing the performance of various MQ channels, including channel name, type, and status.

Best practices for presentation also include these principles:

- Always create custom copies of product-provided workspaces and queries, rather than modifying the original. This will prevent the possibility of customizations being lost due to maintenance, upgrades and migrations. Definition of a naming standard for custom workspaces and queries will make administration and use of OMEGAMON products easier.
- Make copies of product-provided queries and customize them to reduce the amount of data displayed and show that data in the most effective way. Filters and sort fields are some of the options that can be added to custom queries to provide improved display. Unnecessary attributes can be removed from custom queries to reduce data transfer and TEPS workload.

An example of these two practices can be seen in the following screen shots. The Address Space CPU Utilization workspace in the Physical navigator can be difficult to use because many address spaces are running and the default sort order is by ASID. Therefore, the information that interests us most is not always immediately visible.

Tivoli Enterprise Portal Welcome DDS1034 Log out

File Edit View Help

Navigator

View: Physical

- Enterprise
 - Windows Systems
 - z/OS Plexes & Systems
 - More...
 - MVSA
 - More...
 - MVS System
 - DEMOPLX: MVSA: MVSSYS
 - Address Space Overview
- z/VM & Linux Systems
- z/VM Systems

z/DS Sample Navigator Physical

Operations z/US Monitoring

CPU Usage Page: 1 of 4

Address Space CPU Utilization Page: 1 of 4

Job Name	Step Name	Proc Step	SvcClass	SvcClass Period	ASID	JESJOBID	CPU Percent	TCB Percent	SRB Percent	IFA Percent	IFA on CP Percent	zIIP Percent	zIIP on CP Percent	Independent Enclave CPU%	Indi E
MASTER			SYSTEM	1	0X0001	STC02731	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
PCAUTH	PCAUTH		SYSTEM	1	0X0002		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
RASP	RASP		SYSTEM	1	0X0003		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
TRACE	TRACE		SYSTEM	1	0X0004		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
DUMPSRV	DUMPSRV	DUMPSRV	SYSTEM	1	0X0005		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
XCFAS	XCFAS	IEFPROC	SYSTEM	1	0X0006		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
GRS	GRS		SYSTEM	1	0X0007		1.7	1.7	0.0	0.0	0.0	0.0	0.0	0.0	
SMSPDSE	SMSPDSE		SYSTEM	1	0X0008		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
SMSPDSE1	SMSPDSE1		SYSTEM	1	0X0009		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
CTCSVR	CTCSVR	TEFPROC	OPSDEF	1	0X000A		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

For System MVSA

Hub Time: Wed, 01/07/2009 09:11 AM Server Available Address Space CPU Utilization - tivtpeps.demopkg.ibm.com - DDS1034

Applet CMWApplet started

To make this workspace more usable, we will save a custom copy and then begin making changes. First, we do a File, Save Workspace As, to save our custom workspace.

Tivoli Enterprise Portal Welcome DDS1034 Log out IBM

File Edit View Help

Save Workspace Ctrl+S Save Workspace As... F12 Delete Workspace Restore Original Workspace Print Preview... Print Workspace... Ctrl+P Trace Options... Close Exit F3

z/VM & Linux Systems z/VM Systems z/OS Sample Navigator Operations Physical

CPU Usage Page: 1 of 4

Address Space CPU Utilization Page: 1 of 4

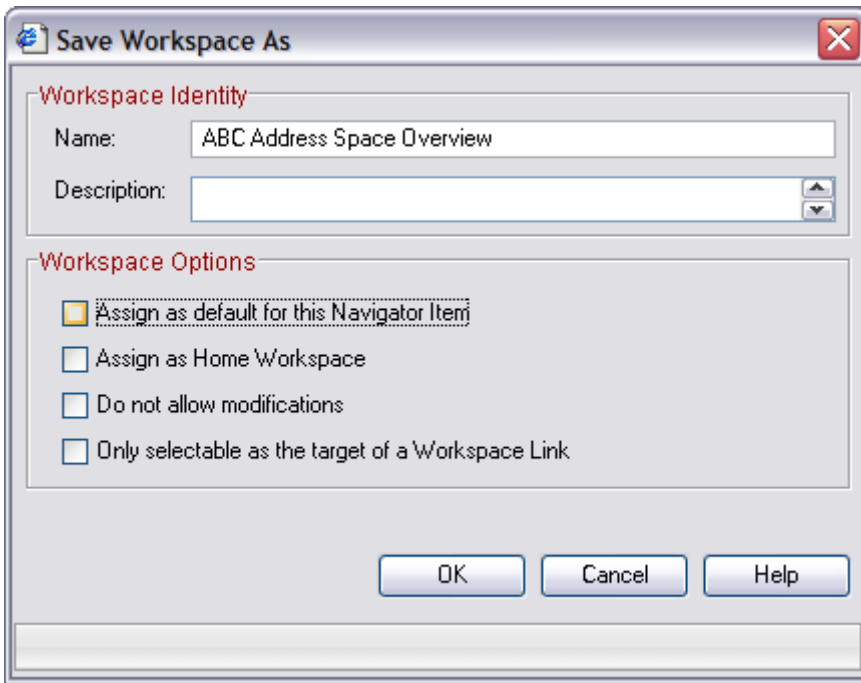
Job Name	Step Name	Proc Step	SvcClass	SvcClass Period	ASID	JESJOBID	CPU Percent	TCB Percent	SRB Percent	IFA Percent	IFA on CP Percent	zIIP Percent	zIIP on CP Percent	Independent Enclave CPU%	Indt E
MASTER			SYSTEM	1	0X0001	STC02731	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
PCAUTH	PCAUTH		SYSTEM	1	0X0002		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
RASP	RASP		SYSTEM	1	0X0003		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
TRACE	TRACE		SYSTEM	1	0X0004		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
DUMPSRV	DUMPSRV	DUMPSRV	SYSTEM	1	0X0005		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
XCFAS	XCFAS	IEFPROC	SYSTEM	1	0X0006		0.4	0.0	0.4	0.0	0.0	0.0	0.0	0.0	
GRS	GRS		SYSTEM	1	0X0007		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
SMSPDSE	SMSPDSE		SYSTEM	1	0X0008		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
SMSPDSE1	SMSPDSE1		SYSTEM	1	0X0009		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
CTCSVR	CTCSVR	TEFPROC	OPSDEF	1	0X000A		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

For System MVSA

Hub Time: Wed, 01/07/2009 10:19 AM Server Available Address Space CPU Utilization - tivtpeps.demopkg.ibm.com - DDS1034

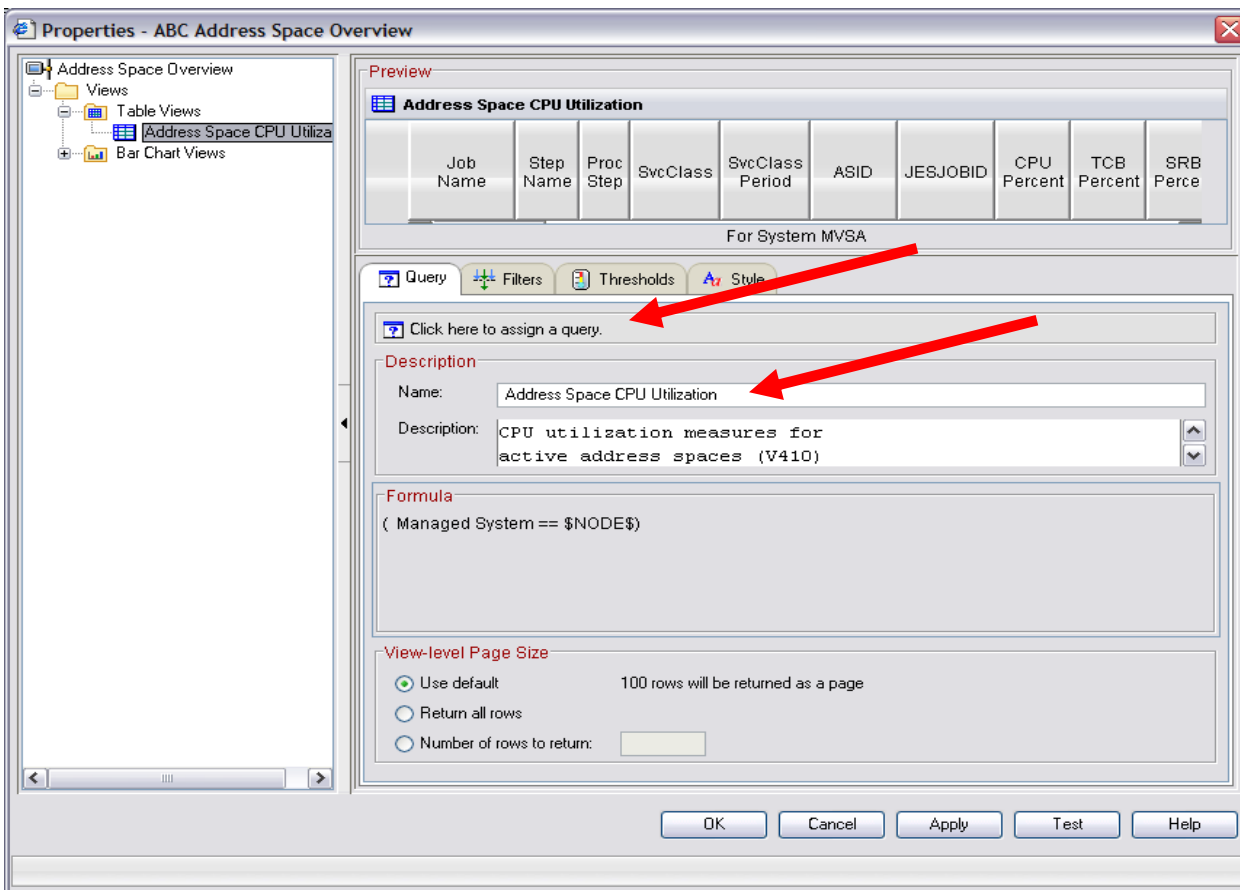
Applet CMWApplet started Internet

Please note in the next screen that the new name contains a prefix of ABC that tells us that this is a custom workspace for company ABC.

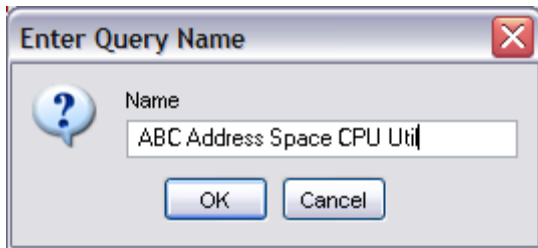
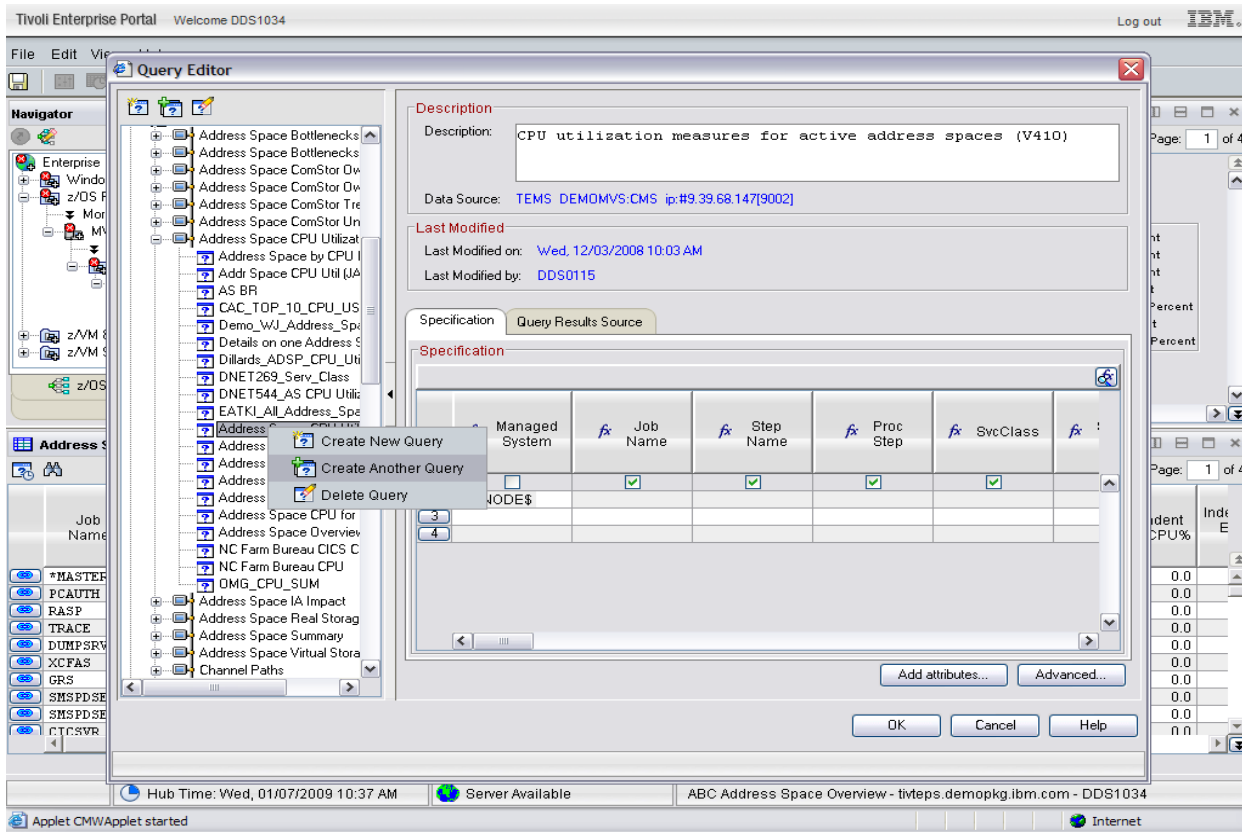


Now that we have saved our custom workspace, we can begin to modify it to make it easier to use.

The first thing we will do is to change the query that is being used to return data to the view in this workspace. We see that our custom workspace is still using a product-provided query, so we will create our own custom copy of the query before making any changes. We will start with the table view at the bottom. By right-clicking within the view and selecting Properties, or by clicking on the “/” symbol in the upper right portion of the view, we open the View Properties dialog box. We can see that the query currently being used is Address Space CPU Utilization.

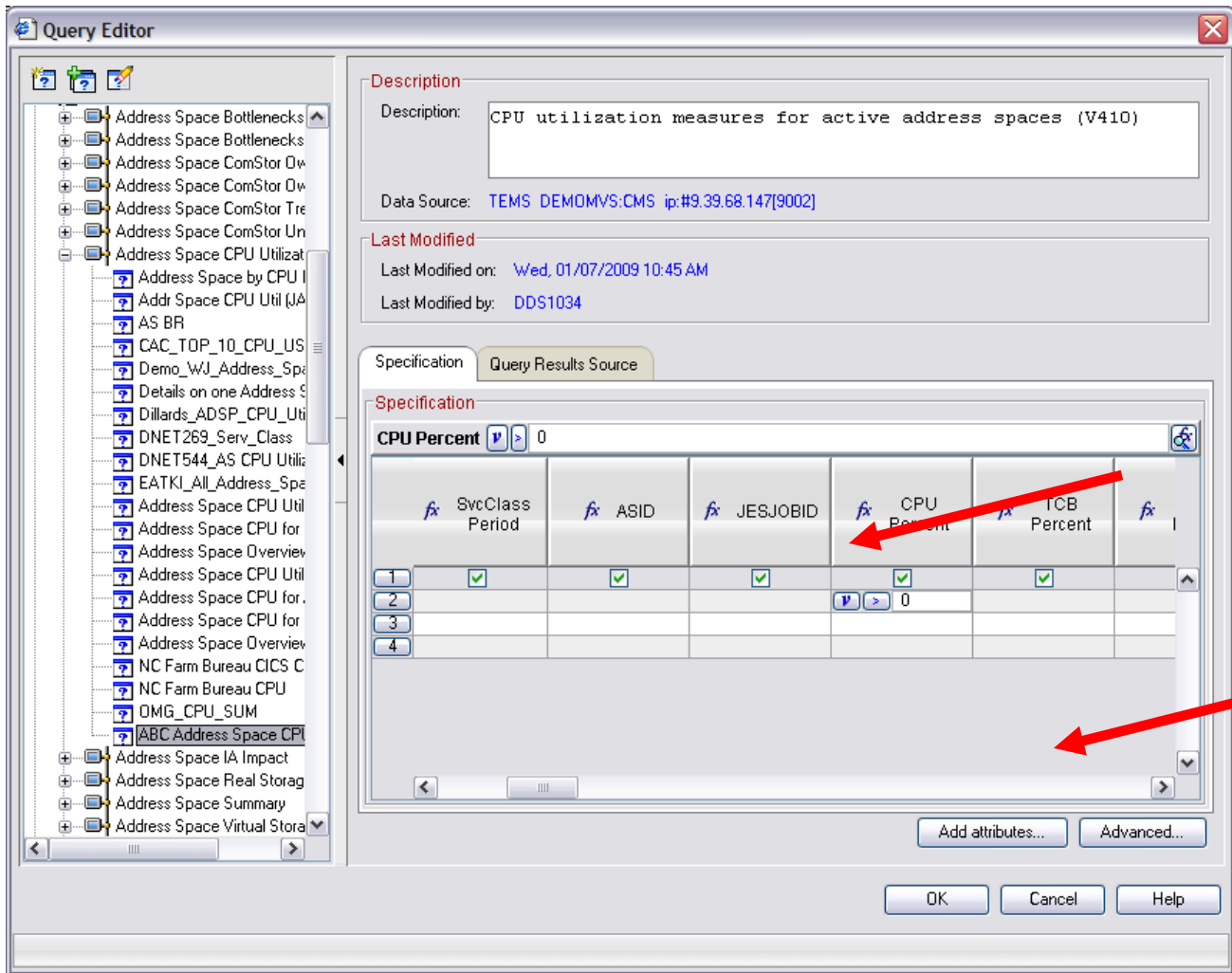


Because we do not want to modify the product-provided query, we will create our own custom copy before making changes. First, we will click on the “Click here to assign a query” button to open the query editor. Then we will create a copy of the query, as shown in the following screen.

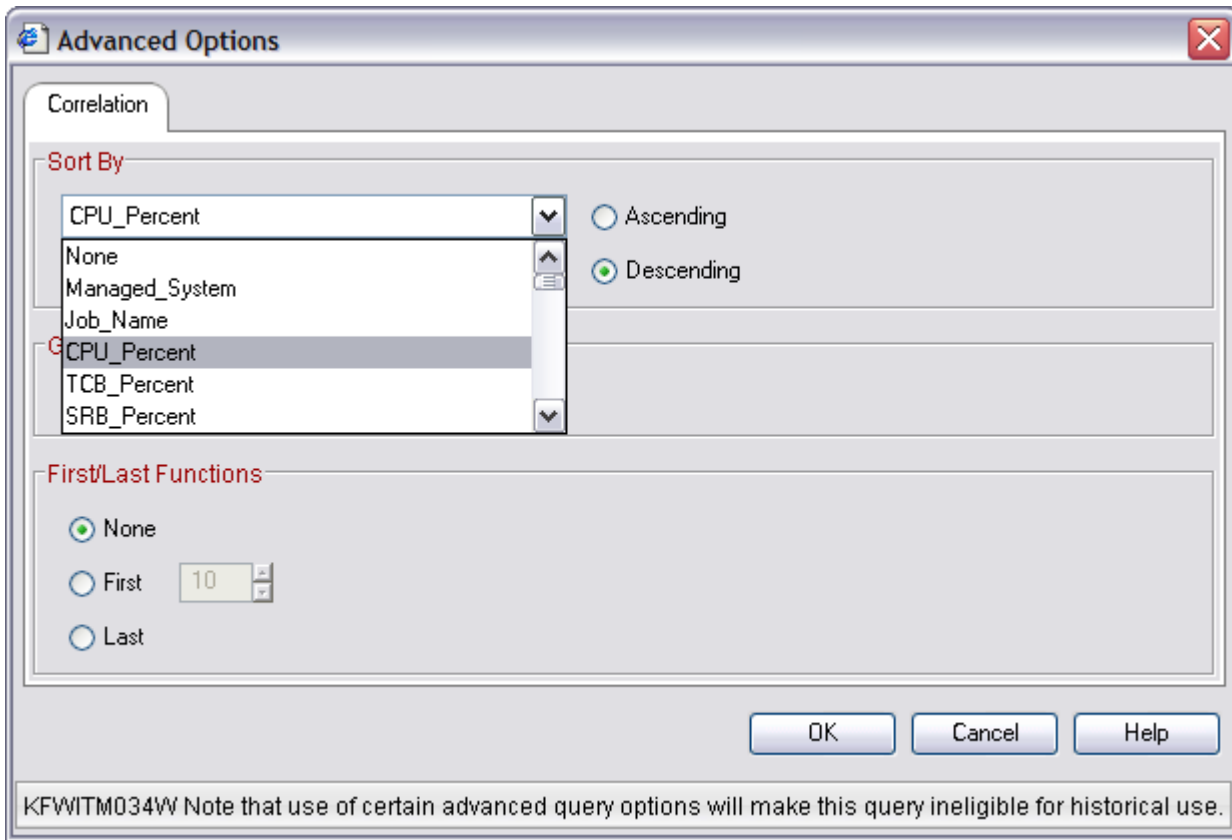


Now that we have our own custom query, we can add a sort order and a filter to manipulate the results that will be returned to the view in our custom workspace.

First, we will add a filter so that tasks using 0% CPU will not be returned. Then we click the “Advanced” button to specify a sort order.



In the Advanced dialog, we select the attribute that we wish to sort by, in this case CPU Percent. We also select Descending sort order so the busiest tasks will be displayed at the top of our table view. Please note that we could also specify First/Last Functions to show the top ten users of CPU, for example, but we will not do that in this exercise.



After selecting OK here, we are returned to the Query Editor dialogue where we will also select OK. Then we are returned to the View Properties dialogue where we will also select OK. The next screen shows our modified workspace.

IBM z Systems **IBM Monitoring – OMEGAMON Best Practices & Performance Guidelines V2**

Tivoli Enterprise Portal Welcome DDS1034 Log out IBM

File Edit View Help

Navigator View: Physical

- Enterprise
 - Windows Systems
 - z/OS Plexes & Systems
 - More...
 - MVSA
 - More...
 - MVS System
 - DEMOPLEX: MVSA: MVSSYS
 - Address Space Overview
 - More...
- z/VM & Linux Systems
- z/VM Systems

z/OS Sample Navigator Physical

Operations

CPU Usage Page: 1 of 4

Address Space CPU Utilization

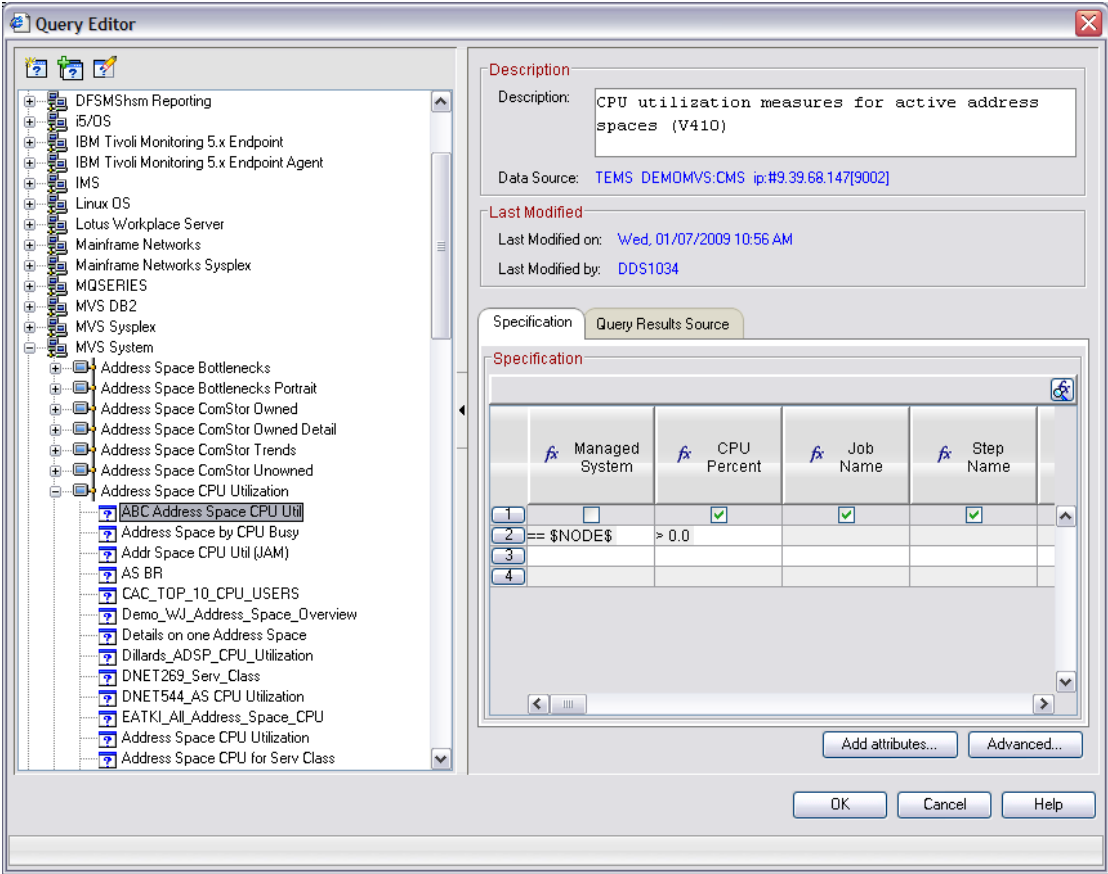
Job Name	Step Name	Proc Step	SvcClass	SvcClass Period	ASID	JESJOBID	CPU Percent	TCB Percent	SRB Percent	IFA Percent	IFA on CP Percent	zIIP Percent	zIIP on CP Percent	Independent Enclave CPU%	Independent Enclave IFA%
NETVIEW	NETVIEW	NETV530	OPSDEF	1	0X0177	STC08725	12.6	12.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0
CXEG02	CXEG02	02CI	OPSDEF	1	0X013A	STC03043	4.3	4.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0
WLM	WLM	IEFPROC	SYSTEM	1	0X000C		1.3	1.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0
XCFAS	XCFAS	IEFPROC	SYSTEM	1	0X0006		1.3	0.0	1.3	0.0	0.0	0.0	0.0	0.0	0.0
RMFGAT	RMFGAT	IEFPROC	SYSSTC	1	0X00DA	STC02903	0.4	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0
CXEGDST	CXEGDST	TEMS	OPSDEF	1	0X0139	STC03039	0.4	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ADHADB1S	ADHADB1S	ADHAGT	OPSDEF	1	0X00D2	STC03680	0.4	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0
CXEGRHA	CXEGRHA	KRHMST00	OPSDEF	1	0X0157	STC03070	0.4	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0
CXEGRNA	CXEGRNA	KRHMST00	OPSDEF	1	0X0138	STC09573	0.4	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0

For System MVSA

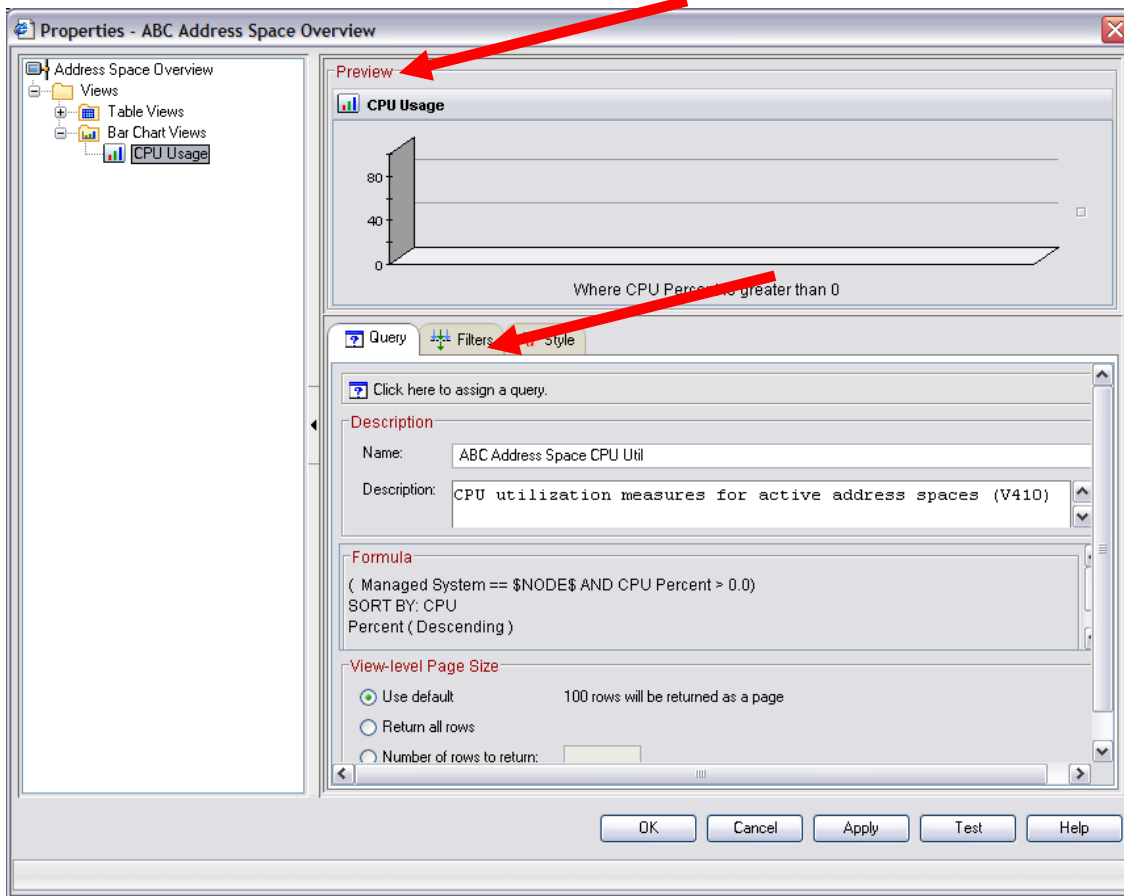
Hub Time: Wed, 01/07/2009 10:51 AM Server Available ABC Address Space Overview - tivtpeps.demopkg.ibm.com - DDS1034

Applet CMWApplet started Internet

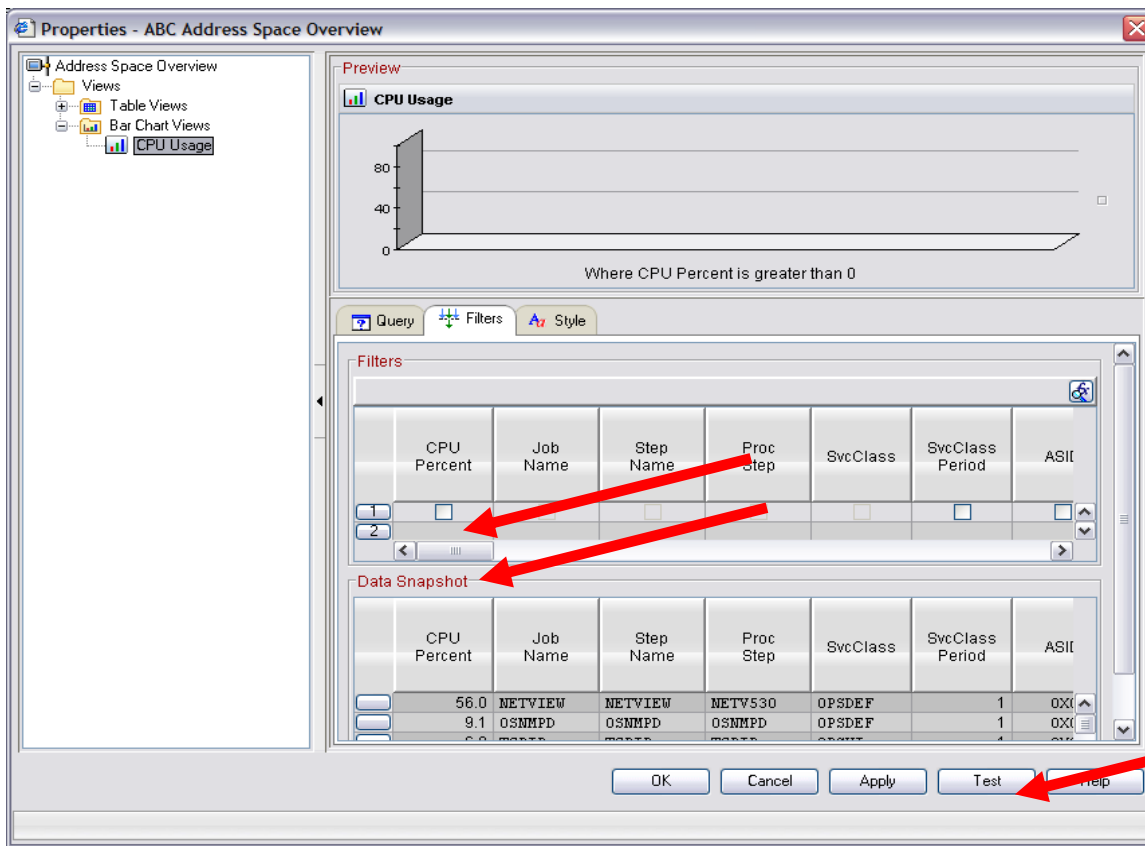
We now have a table view that shows us only the tasks running on the system that truly interest us. But the bar chart view is still using the original query and does not show the same information. Now we can open View Properties for the bar chart view and assign our modified query to that view too. To open View Properties, we will either right-click within the bar chart view, or select the “/” icon in the upper right corner of the bar chart view. Then we will click the “Click here to assign a query” button and select our modified query ABC Address Space CPU Util rather than the default.



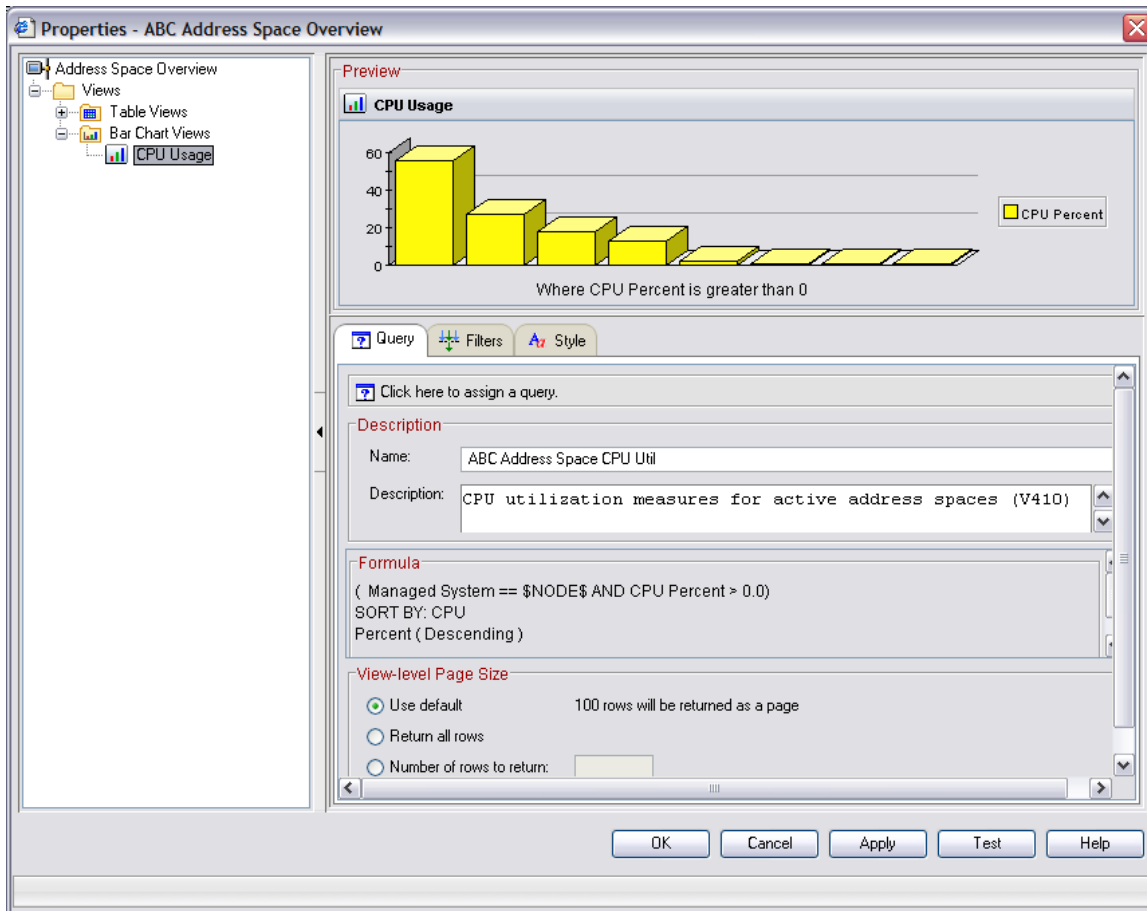
After clicking OK, we are returned to View Properties, where we can see the results of our changes in the Preview window.



Why do we not see any bars in our new bar chart in the Preview window? Let's take a look at the Filters tab to see if we can find out why.



Our Data Snapshot at the bottom clearly shows us that we have data being returned from our query. However, the Filters section shows us that none of the attributes have been selected for charting. This is because we created a new query. By selecting the attributes, we wish to see in the bar chart and then clicking the Test button, we can see our new bar chart in the Preview pane. For this example, we will select the CPU Percent attribute. However, we also could have chosen other attributes such as TCB Percent, SRB Percent, IFA Percent and zIIP Percent if we felt that those metrics were important to show on this bar chart. The next screen shows a preview of our bar chart in View Properties.



Although this is starting to look more usable, there is still something missing. The bars of our new bar chart have no labels to tell us which address spaces are using the CPU. To add labels, we can use the Style tab. Then in the style overview window we can select the Category Axis. When the style window for the Category Axis is displayed, we will select the Category Axis tab. This is shown in the next screen.

Properties - ABC Address Space Overview

Address Space Overview

- Views
 - Table Views
 - Bar Chart Views
 - CPU Usage

Preview

CPU Usage

Where CPU Percent is greater than 0

Query Filters **Style**

Options

- Show Grid
- Hide Category Axis and Value Axis

Category Axis

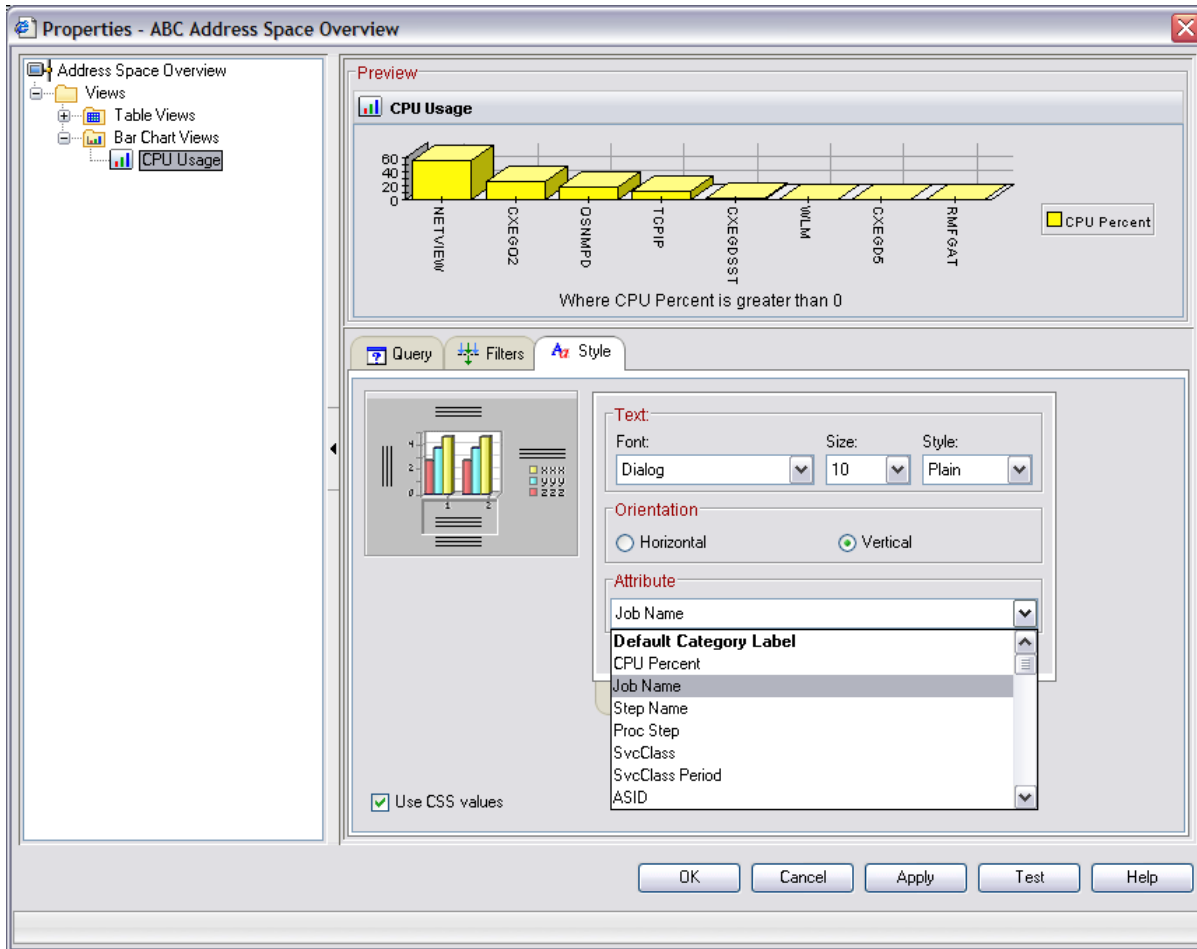
Category Axis - General **Axis Label** **Category Axis**

Use CSS values

OK Cancel Apply Test Help

IBM z Systems **IBM Monitoring – OMEGAMON Best Practices & Performance Guidelines V2**

The Category Axis tab allows us to specify an attribute that will be used as a label for the bars on our bar chart. For this example, we will select Job Name and examine the results.



The Preview window shows us that this will be a useful display of data, so we can save it by clicking the OK button. Our completed, modified workspace now looks like the following screen shot.

Tivoli Enterprise Portal Welcome DDS1034 Log out IBM

File Edit View Help

Navigator

View: Physical

- Enterprise
 - Windows Systems
 - z/OS Plexes & Systems
 - More...
 - MVSA
 - More...
 - MVS System
 - DEMOPLX: MVSA: MVSSYS
 - Address Space Overview
 - More...
 - z/VM & Linux Systems
 - z/VM Systems

z/OS Sample Navigator Physical

Operations

CPU Usage

Where CPU Percent is greater than 0

Address Space CPU Utilization

Job Name	Step Name	Proc Step	SvcClass	SvcClass Period	ASID	JESJOBID	CPU Percent	TCB Percent	SRB Percent	IFA Percent	IFA on CP Percent	zIIP Percent	zIIP on CP Percent	Independent Enclave CPU%	Independent Enclave IFA%
NETVIEW	NETVIEW	NETV530	OPSDEF	1	0X0177	STC08725	56.0	56.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
CXEG02	CXEG02	02CI	OPSDEF	1	0X013A	STC03043	26.9	26.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0
OSNMPD	OSNMPD	OSNMPD	OPSDEF	1	0X0070	STC02810	17.8	17.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0
TCPIP	TCPIP	TCPIP	OPSHI	1	0X0028	STC02736	12.6	12.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0
CXEGDSST	CXEGDSST	TEMS	OPSDEF	1	0X0139	STC03039	2.1	2.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
WLM	WLM	IEFPROC	SYSTEM	1	0X000C		0.8	0.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0
CXEGD5	CXEGD5	AGENT	OPSDEF	1	0X013C	STC09559	0.8	0.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0
RMFGAT	RMFGAT	IEFPROC	SYSSTC	1	0X00DA	STC02903	0.8	0.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0

For System MVSA

Hub Time: Wed, 01/07/2009 11:31 AM Server Available ABC Address Space Overview - tivtaps.demopkg.ibm.com - DDS1034

Applet CMWApplet started Internet

With our changes in place, we now have much quicker access to the information that we really want to see than we did with the original workspace.

IBM z Systems IBM Monitoring – OMEGAMON Best Practices & Performance Guidelines V2

- Consider using dynamic links from high-level overview workspaces within simple custom navigators for quick and easy navigation back into the physical navigator for more detailed monitoring information.

To demonstrate this principle, we will use a very simple high level custom navigator and workspace, and then build a dynamic link into the customized ABC Address Space Overview workspace that we just created. This simple navigator contains a top-level navigator item for an Z/OS Overview and three child navigator items, one for each LPAR in our configuration. For now, we have built only one custom view to show CPU utilization for each LPAR. Now we will build a dynamic link to demonstrate linking to the custom workspace we created earlier.

The screenshot displays the Tivoli Enterprise Portal interface. At the top, it says 'Tivoli Enterprise Portal Welcome DDS1034' and 'Log out IBM'. Below the menu bar, there are several toolbars. The main area is divided into several panes:

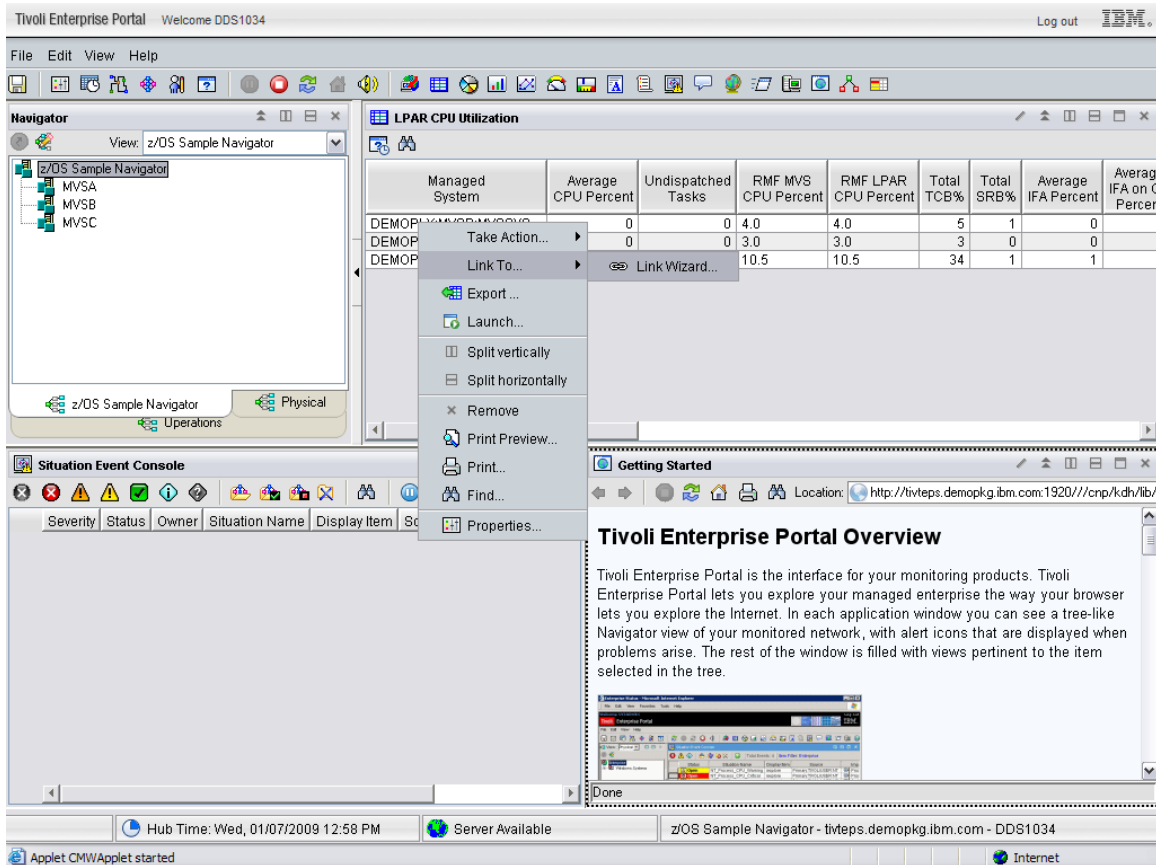
- Navigator:** Shows a tree view under 'z/OS Sample Navigator' with sub-items 'MVSA', 'MVSB', and 'MVSC'. Below the tree are buttons for 'z/OS Sample Navigator', 'Physical', and 'Operations'.
- LPAR CPU Utilization:** A table showing CPU usage for three LPARs.

Managed System	Average CPU Percent	Undispatched Tasks	RMF MVS CPU Percent	RMF LPAR CPU Percent	Total TCB%	Total SRB%	Average IFA Percent	Average IFA on C Percer
DEMOPLX:MVSB:MVSSYS	0	0	4.0	4.0	5	1	0	
DEMOPLX:MVSC:MVSSYS	0	0	3.0	3.0	3	0	0	
DEMOPLX:MVSA:MVSSYS	11	0	10.5	10.5	34	1	1	
- Situation Event Console:** A table with columns: Severity, Status, Owner, Situation Name, Display Item, Source, Impact, Opened. It shows 'Total Events: 0'.
- Getting Started:** A pane titled 'Tivoli Enterprise Portal Overview' with introductory text and a small screenshot of the interface.

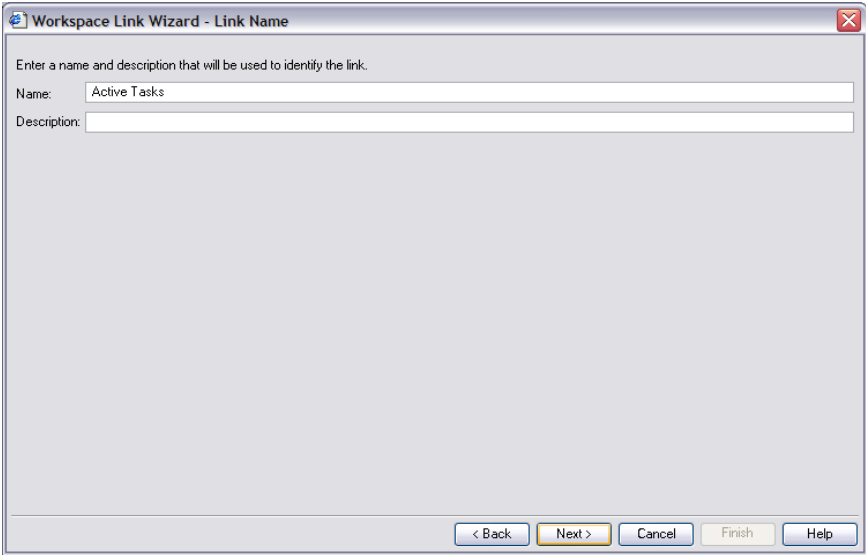
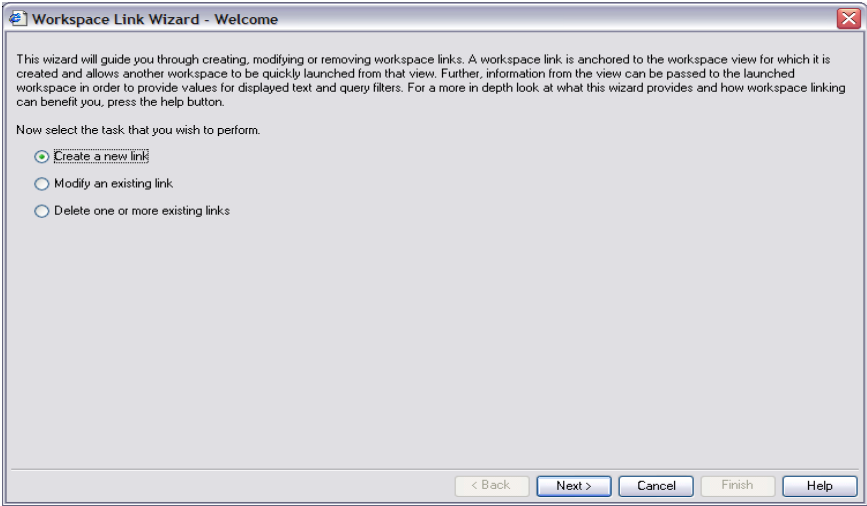
At the bottom, there is a status bar with 'Hub Time: Wed, 01/07/2009 11:51 AM', 'Server Available', and 'z/OS Sample Navigator - tivtpeps.demopkg.ibm.com - DDS1034'. A system tray at the very bottom shows 'Applet CMWApplet started' and 'Internet'.

IBM z Systems IBM Monitoring – OMEGAMON Best Practices & Performance Guidelines V2

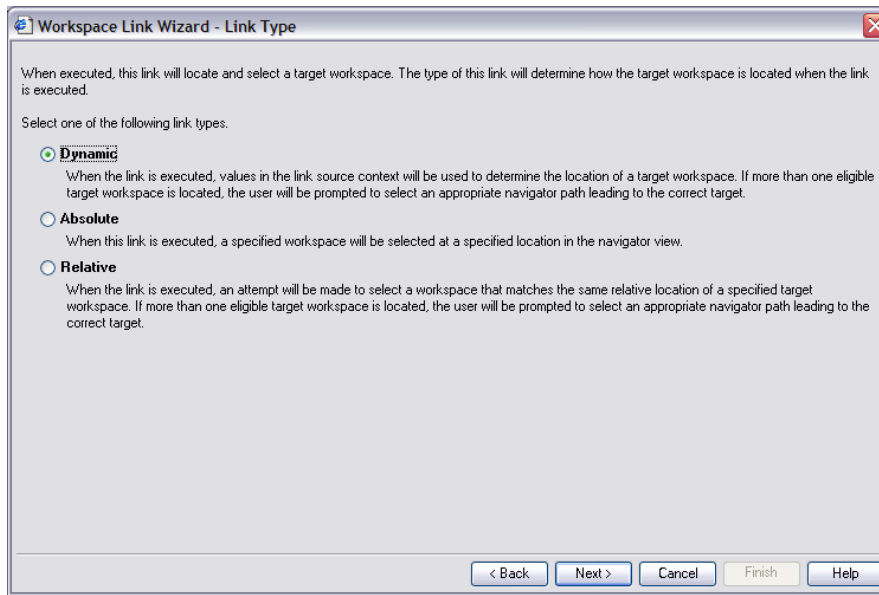
To create a dynamic link, we will start by right-clicking on a row of the table view and selecting Link to, and Link Wizard.



When the Link Wizard opens, we will select Create a New Link, click Next and provide a name of "Active Tasks" for the new link, then click Next again.

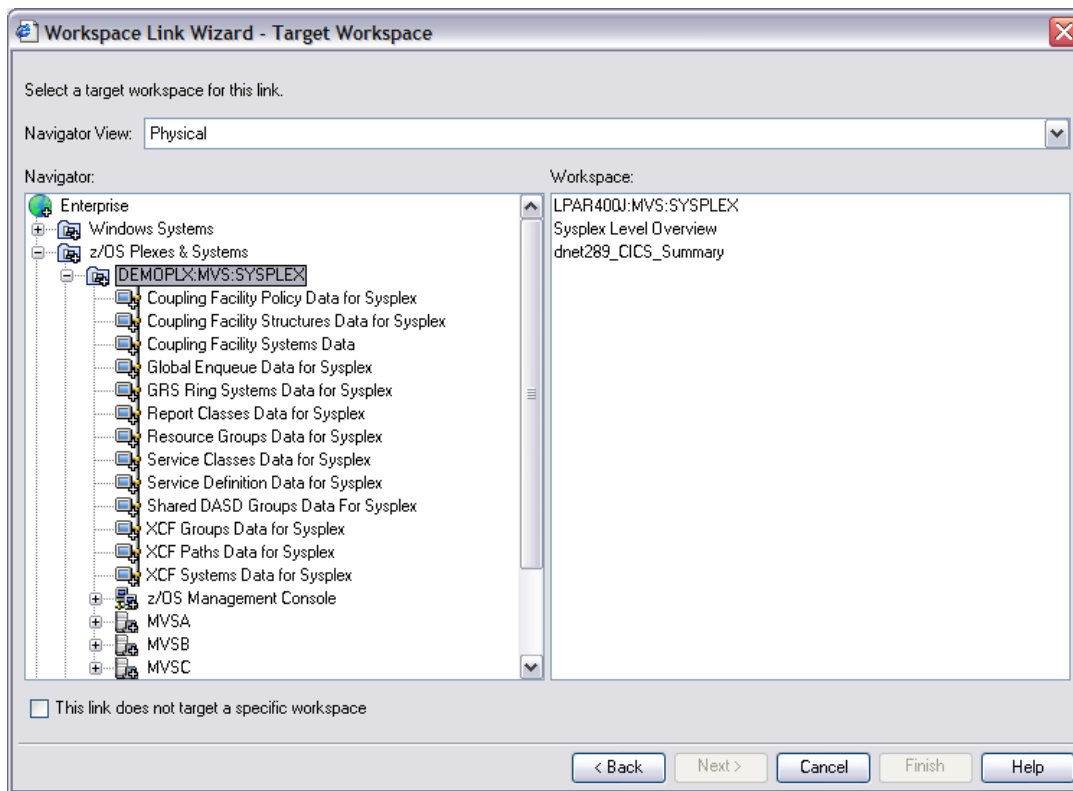


This will provide us with a choice of what type of link we wish to define. Because we want to see active tasks on MVSA when we link from the MVSA row of the table, and likewise for MVSB and MVSC, we want to select a dynamic link, then click next.



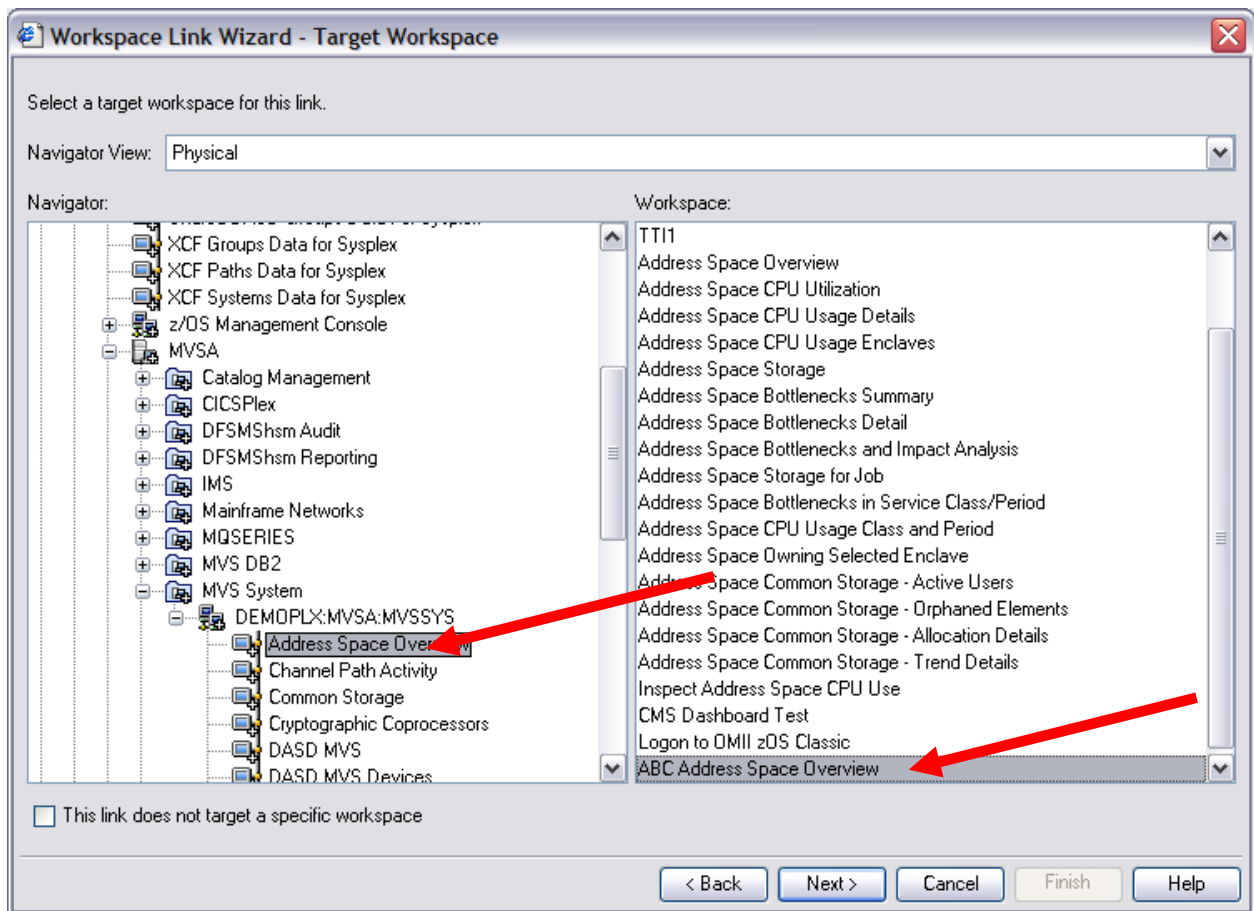
The next screen we see is the Target Workspace selection screen. Here we select which Navigator we would like to link into, which navigator item within that navigator we will link to, and which workspace associated with that navigator item we would like to see.

In the next screen, you can see that we have selected the Physical navigator and expanded the z/OS Plexes & Systems navigator item, then expanded the DEMOPLX:MVS:SYSPLEX navigator item. Please note that systems MVSA, MVSB and MVSC are now visible.



Now we will expand the MVSA navigator item and locate the navigator item and workspace that we wish to link to when this dynamic link is executed. Although we will use the MVSA system for this example of link building, we could choose MVSB or MVSC to build the link and we would get exactly the same results because we are building a dynamic link.

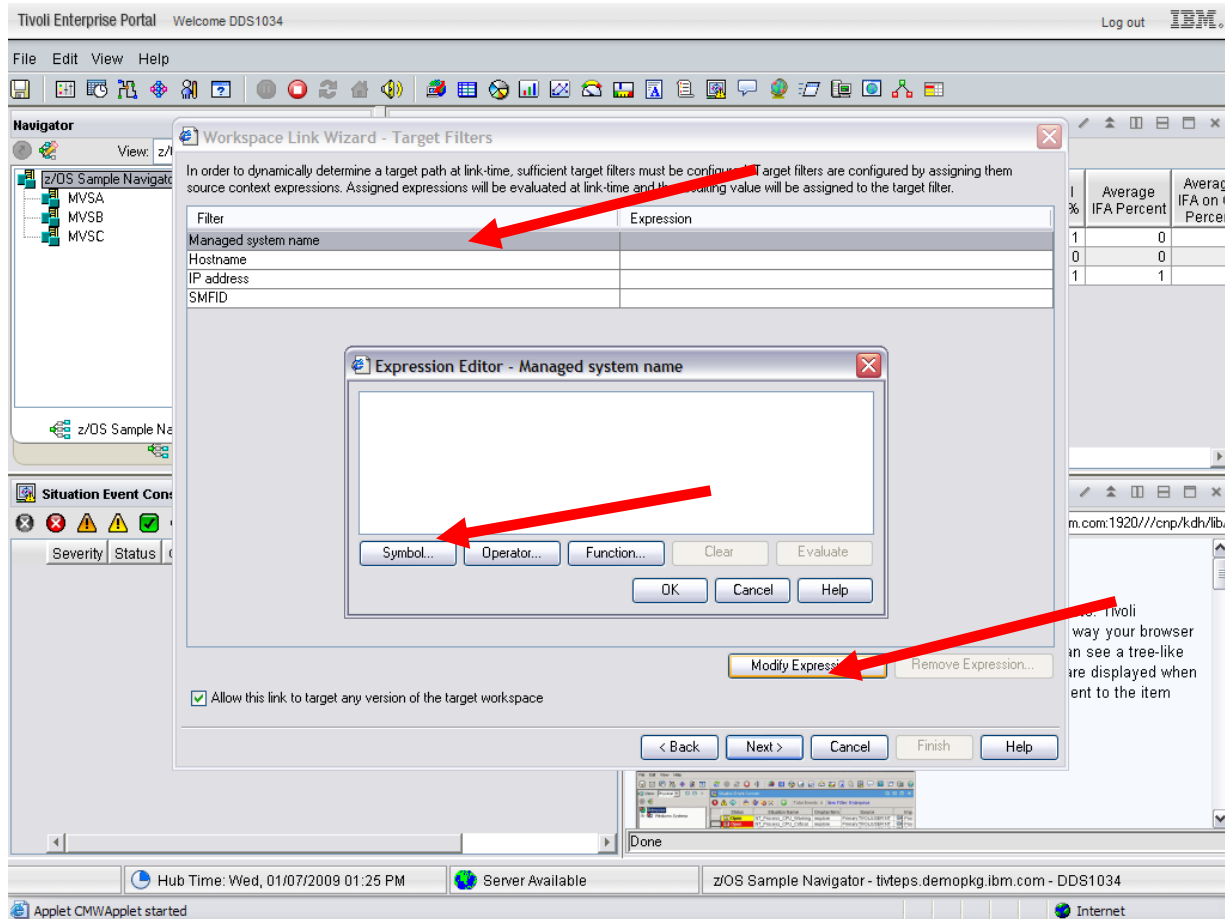
The following screen shot shows that we have expanded the MVSA navigator item, selected and expanded the MVS System agent navigator item, expanded the DEMOPLX:MVSA:MVSSYS managed system navigator item and selected the Address Space Overview navigator item as our link target. In the right panel, we are shown every workspace that is associated with the Address Space Overview navigator item. At the bottom of the list is our newly created workspace, ABC Address Space Overview. We click on that to select it, then click Next.



Because we are defining a dynamic link, the next panel asks us to specify an expression that will be used to determine which version of the ABC Address Space Overview workspace we wish to see. If we link from the MVSA table row, we want to see MVSA data. But if we link from the MVSB or MVSC row, we want to see that specific data instead. The next screens show how we can accomplish that.

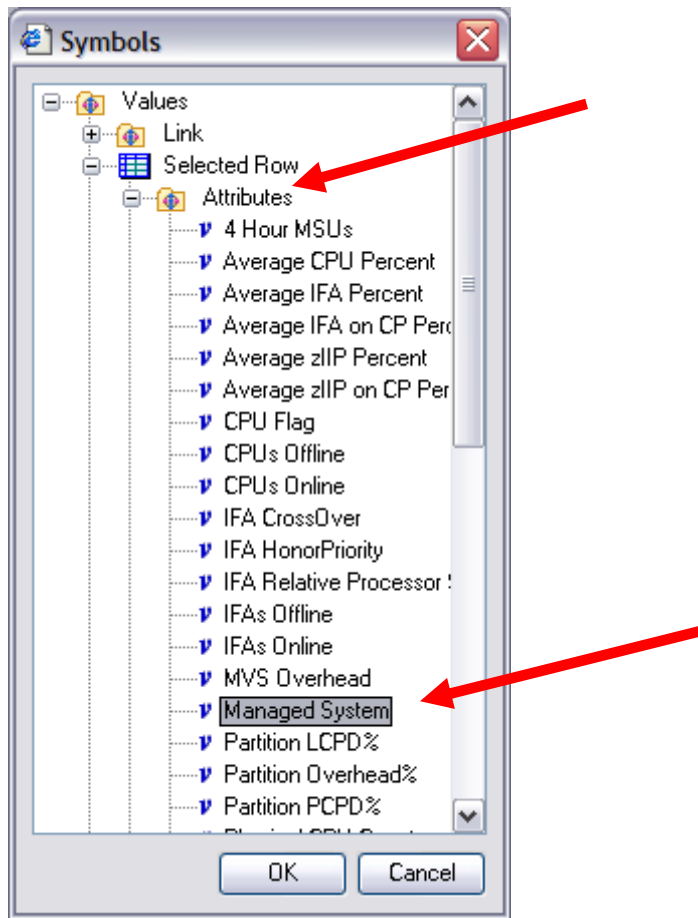
IBM z Systems IBM Monitoring – OMEGAMON Best Practices & Performance Guidelines V2

The Link Wizard gives us four choices of how we will determine which Z/OS system we will link to. For this example, we will use Managed System Name because that information is easily available in the row of data we are linking from. We click on Managed System Name and then on Modify Expression.

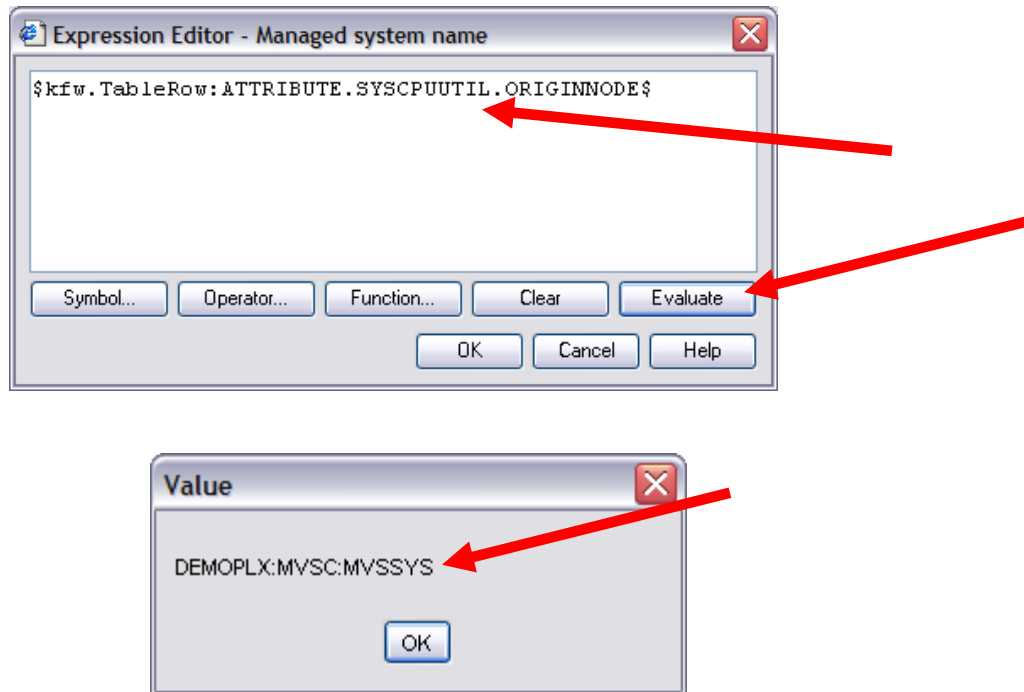


This opens an Expression Editor dialog box where we can assign an attribute that will provide the desired managed system to use for our link. In the Expression Editor, we will click the Symbol button to assign an attribute to our expression.

After clicking Symbol, we are presented with a list of symbols known within the originating workspace. We can choose one of these symbols to build our link. In this case, we will choose Managed System. Note that this symbol is displayed as an Attribute from the Selected Row of the table that we are linking from. Then we click OK.

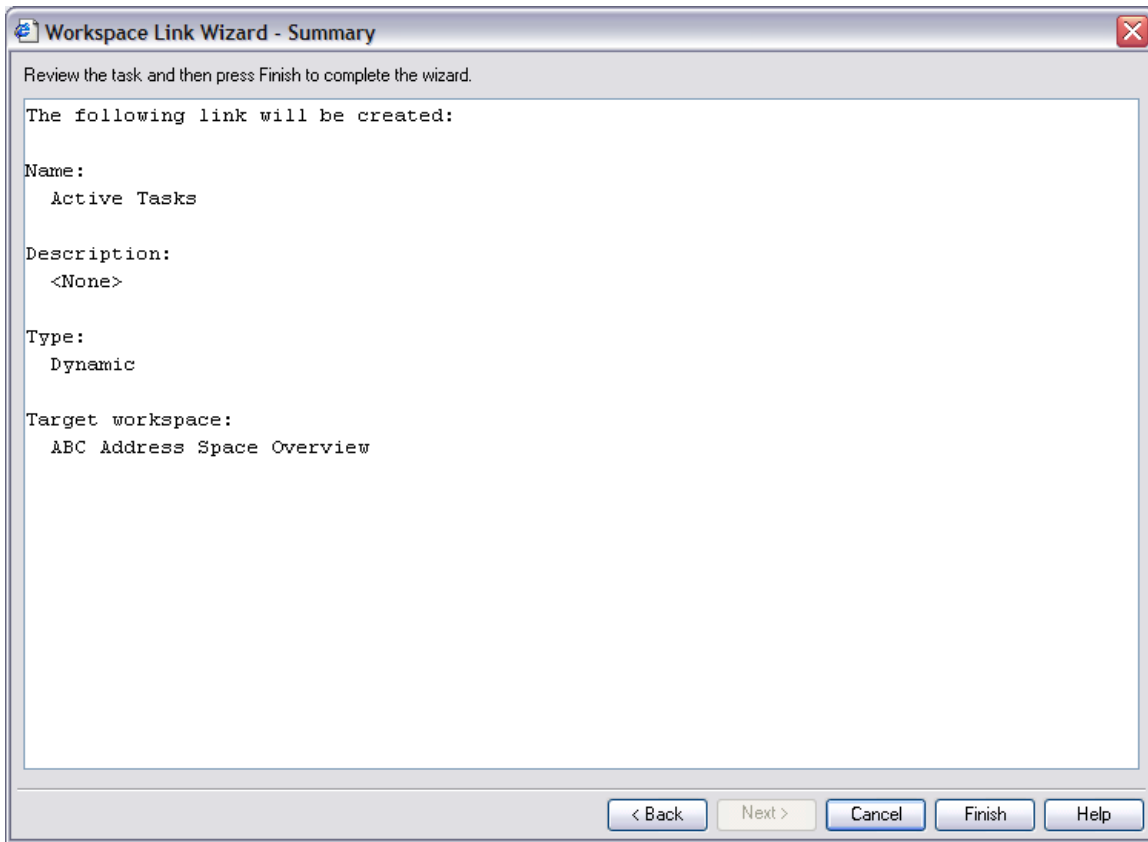


The Expression Editor now shows us the value of the expression that we are building. In this case, it is the attribute name of the managed system in the table row from which we are linking. We should click on the Evaluate button as this point to make sure the expression results will contain the value that we expect.



We see that our expression will result in the managed system name of the Z/OS agent on MVSC. This tells us that we must have started creation of this link from the MVSC row of the table. But since this a dynamic link, it really does not matter. The Link Wizard and the Expression Builder will insure that we will link to the correct target workspace for whichever row of the table we use when we execute the link. We can now click OK a couple of times to return to the Link Wizard and continue.

By clicking Next on the next few screens, we ultimately arrive at a screen where we see a summary of our link definition and can select the Finish button.



When we have clicked Finish, we see that our table in the originating workspace is now populated with link symbols on the left side of the table view.

The screenshot displays the Tivoli Enterprise Portal interface. At the top, there is a menu bar with 'File', 'Edit', 'View', and 'Help'. Below the menu is a toolbar with various icons. The main area is divided into several panes:

- Navigator:** Shows a tree view of the z/OS Sample Navigator with sub-nodes MVSA, MVSB, and MVSC.
- LPAR CPU Utilization:** A table showing CPU utilization for three LPARs. A red arrow points to the first row.
- Situation Event Console:** A pane for monitoring events, currently showing no active events.
- Getting Started:** A pane titled 'Tivoli Enterprise Portal Overview' providing introductory text about the portal.

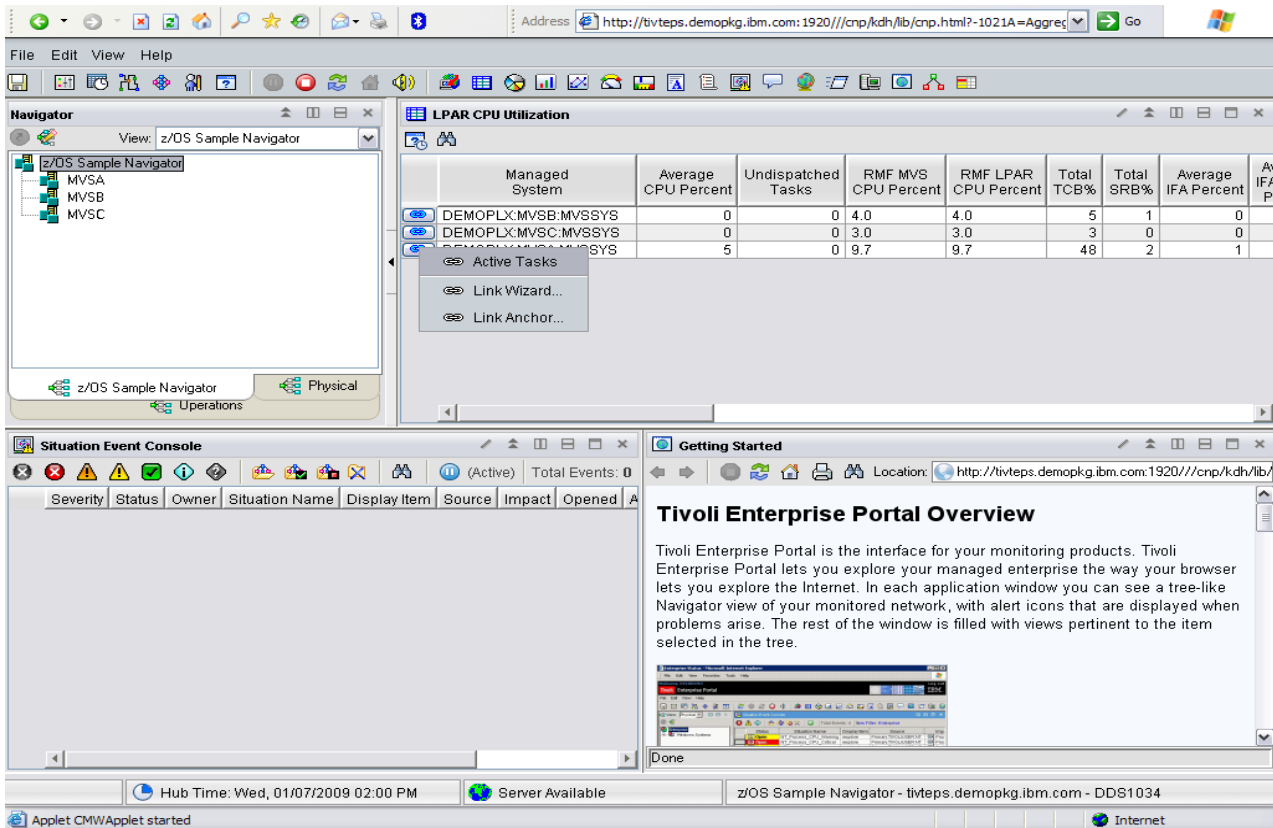
At the bottom of the interface, there is a status bar with information such as 'Hub Time: Wed, 01/07/2009 01:46 PM', 'Server Available', and 'z/OS Sample Navigator - tivteps.demopkg.ibm.com - DDS1034'.

Managed System	Average CPU Percent	Undispatched Tasks	RMF MVS CPU Percent	RMF LPAR CPU Percent	Total TCB%	Total SRB%	Average IFA Percent	AS IF/PS
DEMOPLX:MVSB:MVSSYS	0	0	4.0	4.0	5	1	0	
DEMOPLX:MVSC:MVSSYS	0	0	3.0	3.0	3	0	0	
DEMOPLX:MVSA:MVSSYS	5	0	9.7	9.7	48	2	1	

At this point it is a good idea to save the modified workspace, before executing the link. However, if you do not save the workspace, you will still be asked if you wish to save your changes before the link is executed. To save your workspace, you can click the diskette icon in the TEP tool bar, or select File, Save Workspace from the command bar.

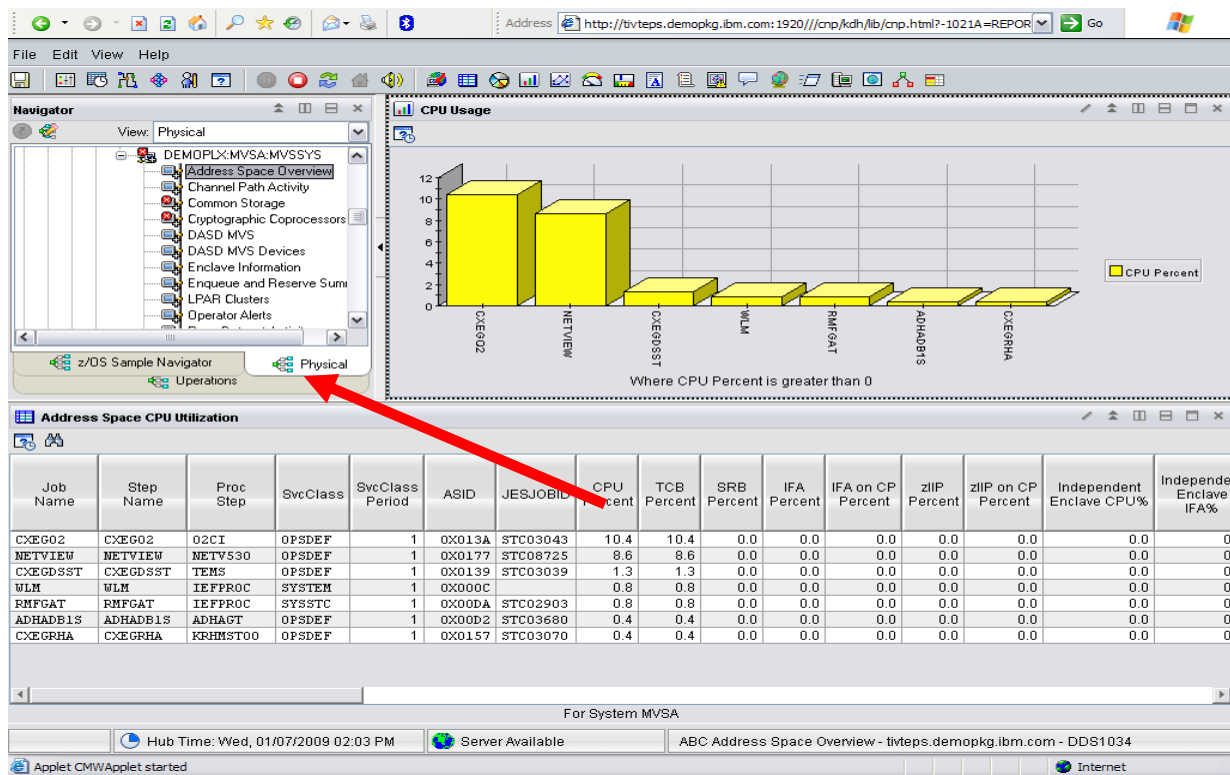
IBM z Systems **IBM Monitoring – OMEGAMON Best Practices & Performance Guidelines V2**

To demonstrate the new link, we can either left-click on the link icon in one of the table rows to take the default link, or we can right-click to see what choices we have for available link destinations. In the next screen shot, we right-click and select our new link from the MVSA row of the table. We can see that Active Tasks is currently the only link that we have defined. If we wished to define additional links to enable navigation to different target workspaces, we could easily do so using the same process that we used to build the Active Tasks link.



When we right-click on the link icon and then click the Active Tasks link, we are taken to the specified link location as shown in the next screen.

IBM z Systems IBM Monitoring – OMEGAMON Best Practices & Performance Guidelines V2



Note that we are now in the Physical navigator, looking at MVSA data in our customized ABC Address Space Overview workspace. If we had linked from the MVSB or MVSC row of the table, we would see this same workspace populated with the correct data for the MVSB or MVSC LPAR.

This represents just one example of the ways in which simple navigators and overview workspaces can be used to create different views of the monitoring environment for various monitoring audiences. This also demonstrates the concept of using dynamic links to navigate from simple overview workspaces back into the Physical navigator to leverage the extensive monitoring information that is available there.

18.2 Alerting Best Practices

Several overall alerting guidelines should be utilized, regardless of the OMEGAMON product being used or the environment being monitored. Each of these techniques will be discussed in greater depth in the following sections.

- Examine the product-provided situations and determine which of them should be used to provide alerting for your unique monitoring needs. Always create custom copies of the product-provided situations, rather than modifying the original. This will prevent the possibility of customizations being lost due to maintenance, upgrades and migrations. Definition of a naming standard for custom situations will make administration and use of OMEGAMON products easier. Product-provided situations that are enabled by default can be disabled after a custom copy is created, or if the situations are not required in your environment, to prevent unnecessary monitoring overhead.

The screenshot displays the Tivoli Enterprise Portal interface. At the top, it says 'Tivoli Enterprise Portal Welcome DDS1034' and 'Log out IBM'. Below the menu bar, there is a toolbar with various icons. The main area is divided into several panes:

- Navigator:** Shows a tree view of the z/OS Sample Navigator with sub-items MVSA, MVSB, and MVSC.
- LPAR CPU Utilization:** A table showing CPU utilization for three managed systems. A red arrow points to the 'Average CPU Percent' column header.
- Situation Event Console:** A pane with a toolbar and a table header for monitoring events.
- Getting Started:** A pane with a title 'Tivoli Enterprise Portal Overview' and introductory text.

The LPAR CPU Utilization table data is as follows:

Managed System	Average CPU Percent	Undispatched Tasks	RMF MVS CPU Percent	RMF LPAR CPU Percent	Total TCB%	Total SRB%	Average IFA Percent	Av IFA P
DEMOPLX:MVSB:MVSSYS	0	0	3.6	3.6	4	1	0	
DEMOPLX:MVSC:MVSSYS	0	0	3.3	3.3	3	0	0	
DEMOPLX:MVSA:MVSSYS	4	0	8.5	8.5	25	2	1	

The Situation Event Console table header includes: Severity, Status, Owner, Situation Name, Display Item, Source, Impact, and Opened.

The Getting Started pane contains the following text:

Tivoli Enterprise Portal Overview

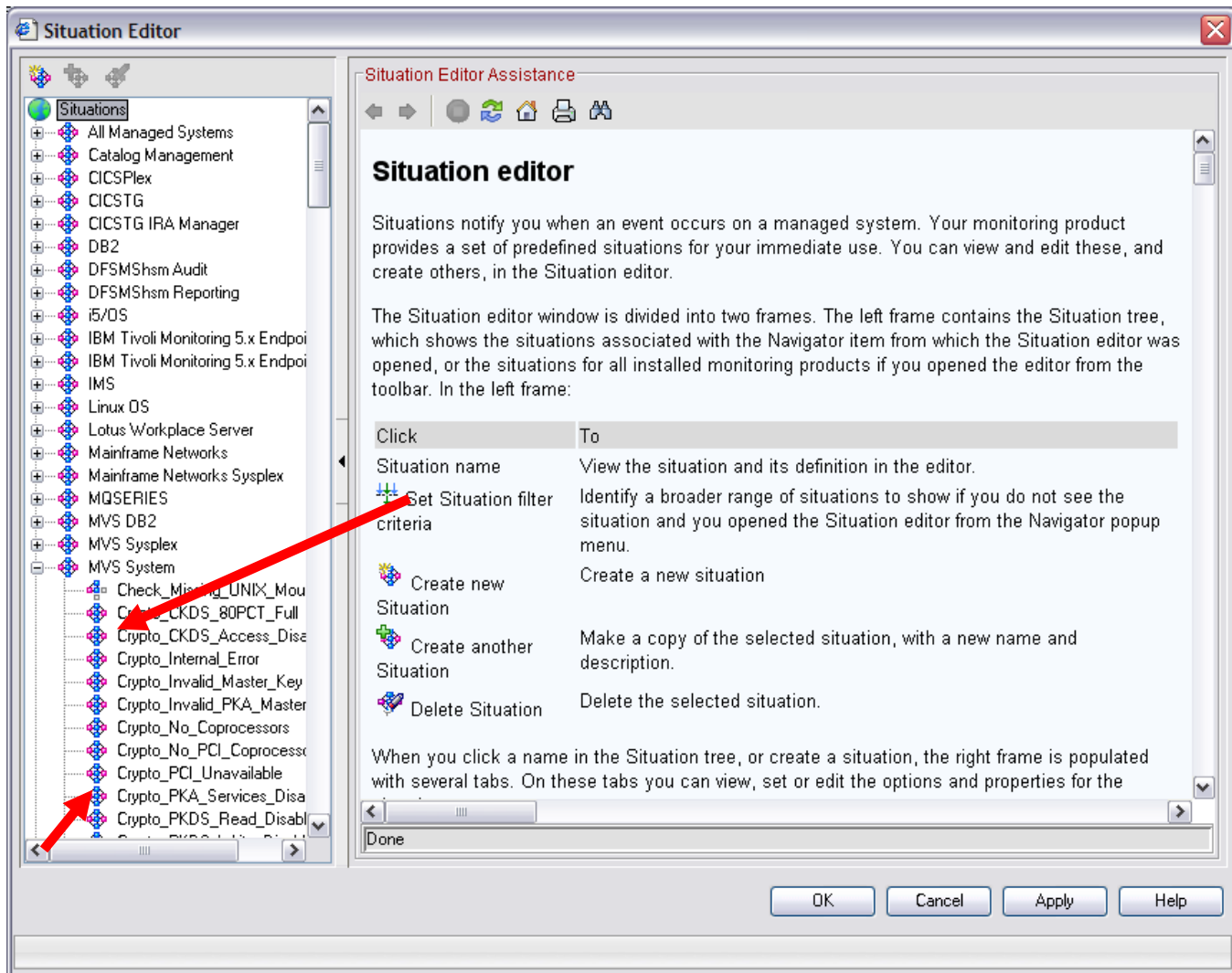
Tivoli Enterprise Portal is the interface for your monitoring products. Tivoli Enterprise Portal lets you explore your managed enterprise the way your browser lets you explore the Internet. In each application window you can see a tree-like navigator view of your monitored network, with alert icons that are displayed when problems arise. The rest of the window is filled with views pertinent to the item selected in the tree.

At the bottom of the interface, there is a status bar showing 'Hub Time: Fri, 01/09/2009 01:57 PM', 'Server Available', and 'z/OS Sample Navigator - tivteps.demopkg.ibm.com - DDS1034'. There is also a small 'Applet CMWApplet started' notification and an 'Internet' icon.

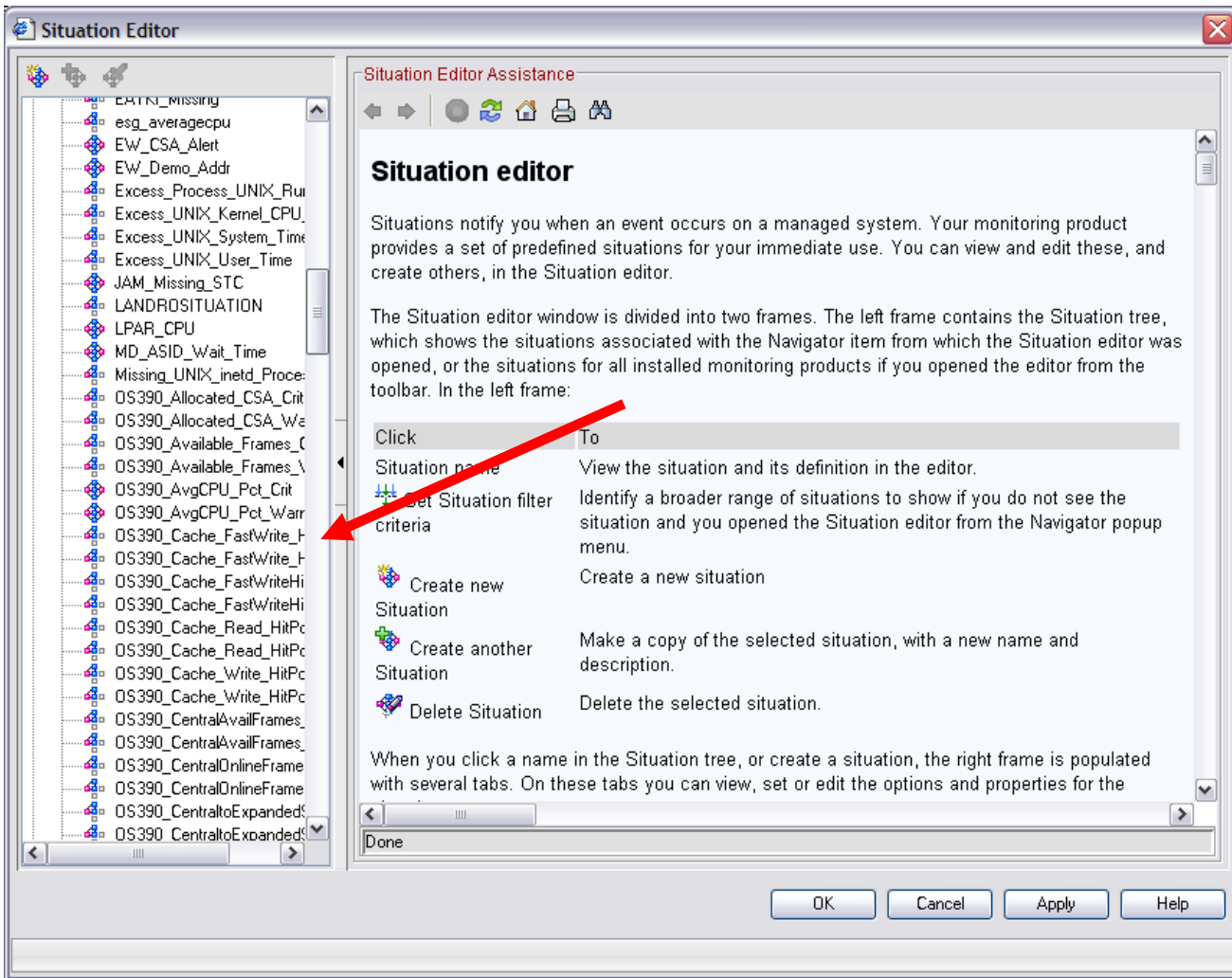
Clicking on the situation editor icon, above, will open the situation editor window as shown in the next screen.

IBM z Systems IBM Monitoring – OMEGAMON Best Practices & Performance Guidelines V2

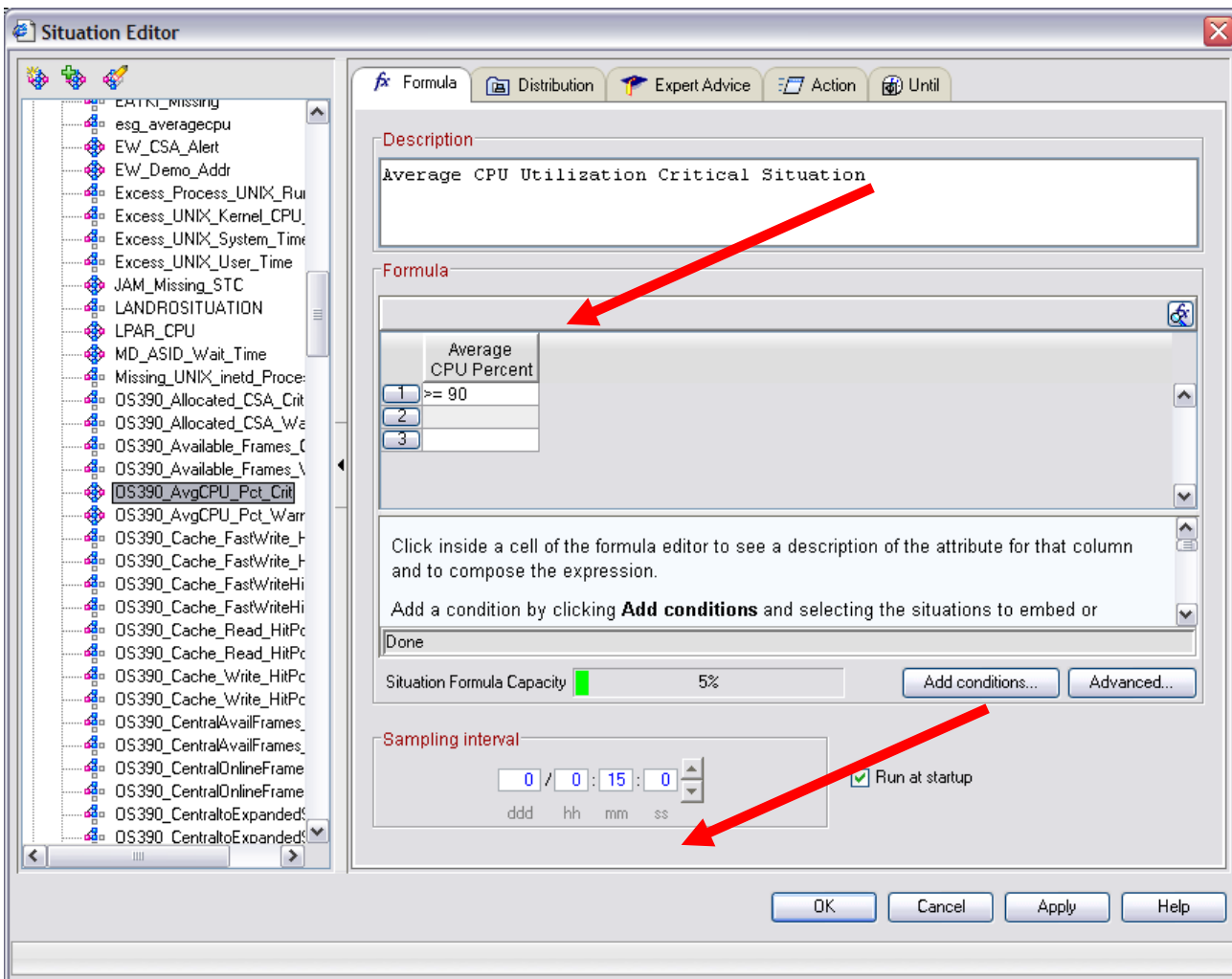
The situation editor contains one navigator item for each product for which support has been installed. For this example, we have expanded the Z/OS Systems navigator item so we can examine existing Z/OS situations. Please note that some of the situation icons are partially grayed out, as shown by the longer arrow. This indicates situations that have not been distributed to run on any agents. The shorter arrow shows what a situation icon looks like for situations that have been distributed to one or more agents.



As we scroll down through the list of situations, we can see by the names of some of the situations that several custom situations have already been created. For this exercise, we will concentrate on those situations that start with OS390, since they account for many of the product-provided situations for the OMEGAMON on z/OS product.



For this example, let's examine the OS390_AvgCPU_Pct_Crit situation and see how we may wish to deploy a copy of that situation into our monitoring environment. By clicking on the OS390_AvgCPU_Pct_Crit situation name, we can open that situation in the situation editor.



We see that this situation has been designed to monitor for 90% or greater utilization of a z/OS LPAR. We can also see that this situation will check with the agent, or TEMA, every 15 minutes to determine whether the condition is true. If we determine that CPU utilization of 90% or greater is a condition that we wish to be alerted about, then we should proceed to make our own, custom copy of this situation. To do that, we can right click the situation name in the navigator and select Create Another to make a copy of the product provided situation. We could also have chosen the Create Another icon from the tool bar, which is pointed out in the next screen.

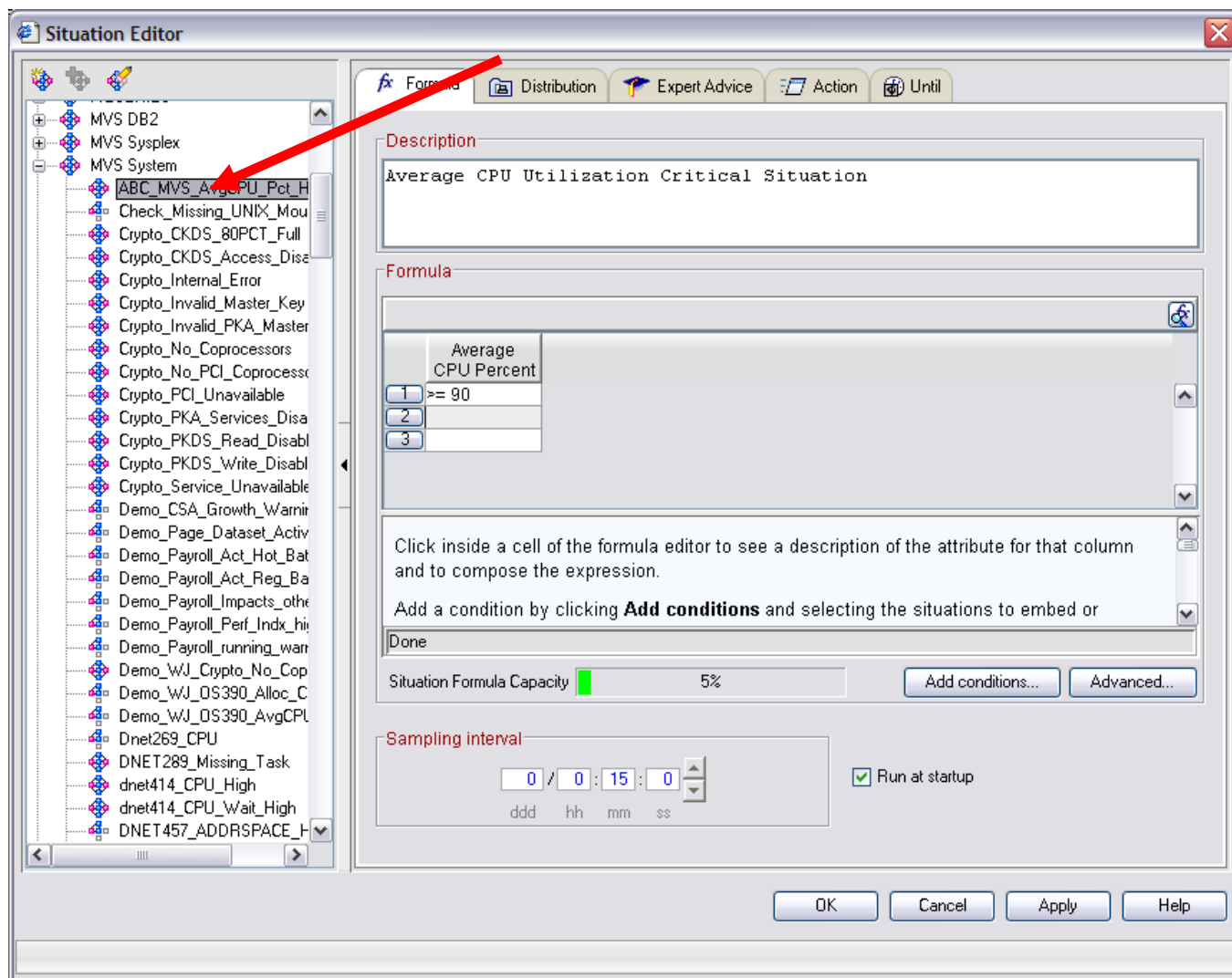
The screenshot shows the 'Situation Editor' window in the Tivoli Enterprise Portal. A red arrow points to the 'Create Another...' option in the context menu for the 'OS390_AverageCPU' situation. The main window displays the 'Formula' tab for a situation named 'Average CPU Utilization Critical Situation'. The formula is 'Average CPU Percent >= 90'. The 'Sampling interval' is set to 0:0:15:0 (dd:hh:mm:ss) and 'Run at startup' is checked. The 'Severity' is set to 'Critical'. The 'Description' field contains 'Average CPU Utilization Critical Situation'. The 'Formula' field contains 'Average CPU Percent >= 90'. The 'Sampling interval' is set to 0:0:15:0. The 'Run at startup' checkbox is checked. The 'Add conditions...' and 'Advanced...' buttons are visible. The 'OK', 'Cancel', 'Apply', and 'Help' buttons are at the bottom.

After using Create Another, we are prompted for a new situation name, which we will specify as ABC_MVS_AvgCPU_Pct_High. The ABC is used to tell us that this is a custom situation for company ABC. The MVS is added to let us know that this situation is written for the Z/OS monitoring agent. Avoiding names that include the words “Warn” or “Crit” gives us the flexibility to assign varying severities to our new situation in the future, depending on our requirements.

The 'Create Situation' dialog box is shown. The 'Name' field contains 'ABC_MVS_AvgCPU_Pct_High'. The 'Description' field is empty. The 'Monitored Application' is set to 'MVS System'. The 'Correlate Situations across Managed Systems' checkbox is unchecked. The dialog box includes instructions for naming conventions: 'Situation name: 1) Must be 31 characters or less, 2) Must start with an alphabetic character (a-z, A-Z), 3) May contain any alphabetic, numeric (0-9) or underscore (_) character, 4) Must end with an alphabetic or numeric character.' The 'OK', 'Cancel', and 'Help' buttons are at the bottom.



After creating our new situation, we can proceed with making changes that will customize the situation for use in our monitoring environment. Please note that in the following screen, we are now changing the situation named ABC_MVS_AvgCPU_Pct_High, rather than the product-provided situation.



Before saving this situation, some of the other options that we may wish to examine include the Distribution, Expert Advice and Action tabs. We may also wish to consider whether the evaluation interval of 15 minutes is reasonable for what we are trying to monitor. More frequent evaluations will drive additional work by the agent to retrieve the data. Less frequent evaluations will potentially result in less timely notification when a problem occurs. For each situation that is created, the benefits of quick notification should be weighed against the cost of more frequent evaluation to determine an optimal interval. It may be desirable to evaluate situations for very critical problems more frequently, while evaluating situations for less critical problems at longer intervals.

For more information about customizing situations, please refer to the [IBM Monitoring User's Guide](#).

A current version of this manual is located here:

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

- All running situations should have some type of action defined, either automated or manual, that will be taken when the situation is true and an event opens. Events that occur with no action taken usually represent pure, wasted overhead.

When we created our custom situation, `ABC_MVS_AvgCPU_Pct_High`, the first thing we should have asked ourselves was “when this situation becomes true and an event is opened, what action will be taken?” If the answer to that question is “no action will be taken” then it is unlikely that this situation is really required.

Actions can be divided into two general categories: manual actions and automated actions.

An example of a manual action would be to have the group responsible for systems monitoring using ITM and OMEGAMON watch for open events and react to those open events by performing a predefined action. This action may be specified in a procedures document, or it could be included in the Expert Advice section of each customized situation.

Before we can respond to an open event, we must first make sure that the situation has been associated with a navigator item. To demonstrate this, we can associate our new situation with our simple custom navigator as shown in the following screen shots. To ensure that an event is produced, the situation has been modified to look for CPU busy of 3% or greater.

First, we click on the navigator item where we wish to associate the situation. In this case, z/OS Sample Navigator. Then we click on Situations . . . as show in the next screen.

The screenshot displays the Tivoli Enterprise Portal interface. At the top, it shows 'Tivoli Enterprise Portal Welcome DDS1034' and a 'Log out' button. Below the menu bar, the 'Navigator' pane on the left shows a tree view of 'z/OS Sample Navigator' with a context menu open. A red arrow points to the 'Situations...' option in this menu. The main pane displays 'LPAR CPU Utilization' with the following table:

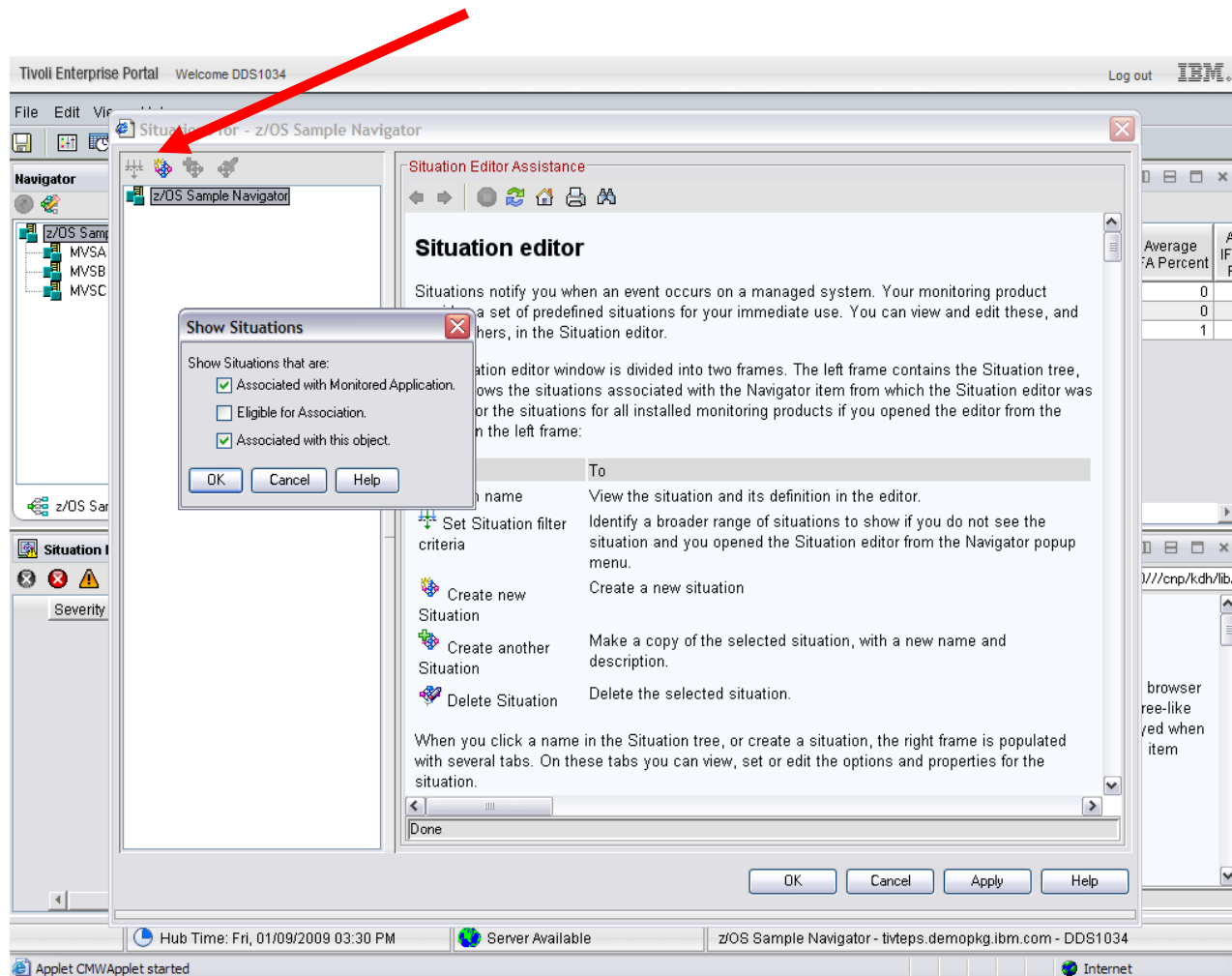
Managed System	Average CPU Percent	Undispatched Tasks	RMF MVS CPU Percent	RMF LPAR CPU Percent	Total TCB%	Total SRB%	Average IFA Percent	AIF/P
DEMOPLX:MVSB:MVSSYS	0	0	3.6	3.6	4	1	0	
DEMOPLX:MVSC:MVSSYS	0	0	3.3	3.3	3	0	0	
DEMOPLX:MVSA:MVSSYS	4	0	8.5	8.5	25	2	1	

Below the table, there is a 'Situation Event Console' and a 'Getting Started' pane. The 'Getting Started' pane contains the text: 'Tivoli Enterprise Portal Overview. Tivoli Enterprise Portal is the interface for your monitoring products. Tivoli Enterprise Portal lets you explore your managed enterprise the way your browser lets you explore the Internet. In each application window you can see a tree-like Navigator view of your monitored network, with alert icons that are displayed when problems arise. The rest of the window is filled with views pertinent to the item selected in the tree.'

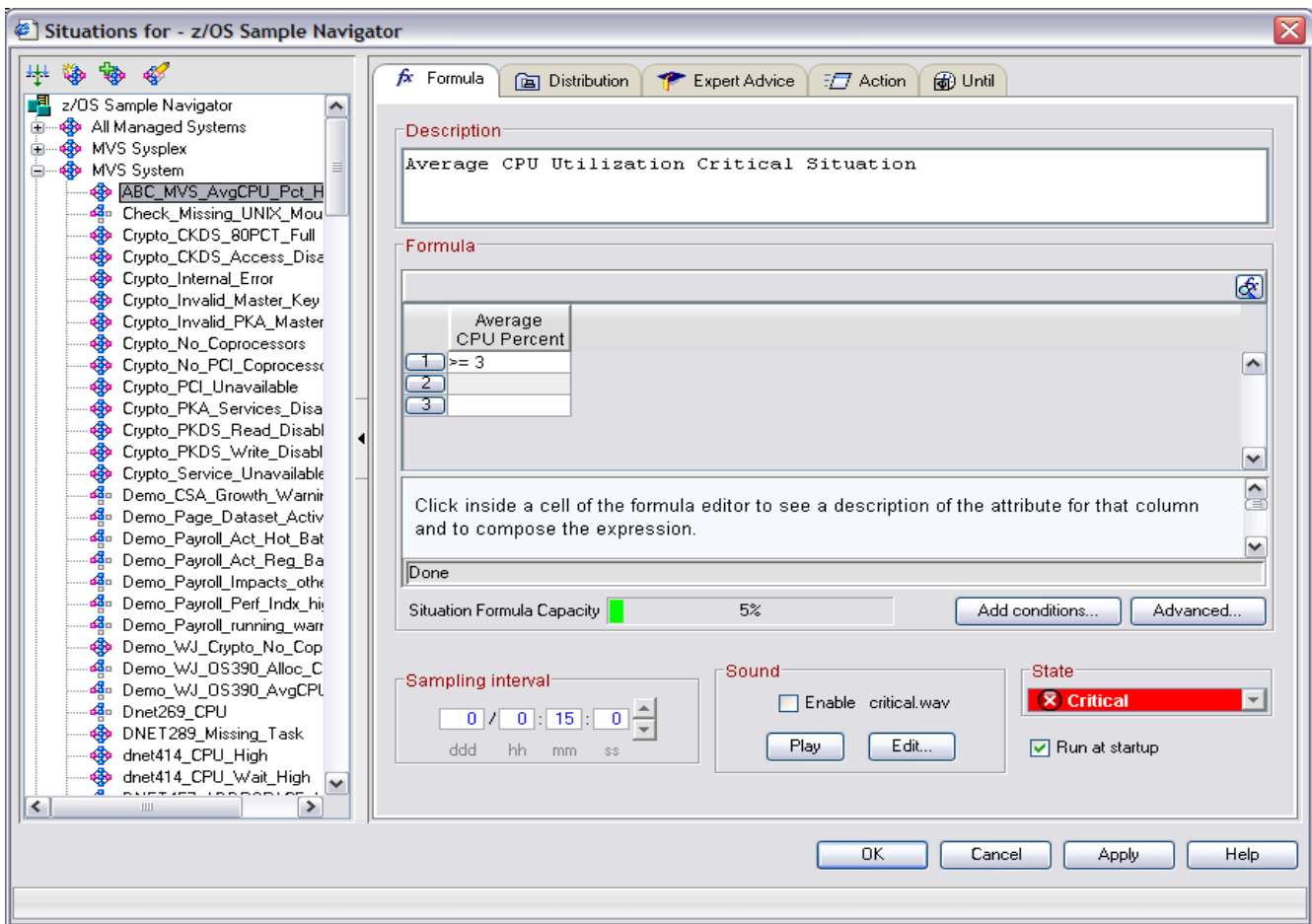
At the bottom of the interface, there is a status bar with the following information: 'Hub Time: Fri, 01/09/2009 03:24 PM', 'Server Available', 'z/OS Sample Navigator - tivtps.demopkg.ibm.com - DDS1034', and 'Applet CMWApplet started'.

IBM z Systems IBM Monitoring – OMEGAMON Best Practices & Performance Guidelines V2

This causes the situation editor to open for the navigator item that we have clicked. In this case, there are currently no situations associated with the z/OS Sample Navigator item, so there are no situations in our list. To find the situation(s) that we wish to associate with the navigator item, we can use the Set situation filter criteria . . . tool as show in the next screen. Selecting “Associated with Monitored Application” in addition to the default of “Associated with this object” will provide us with a list of all situations that have been defined for the z/OS monitoring agent.

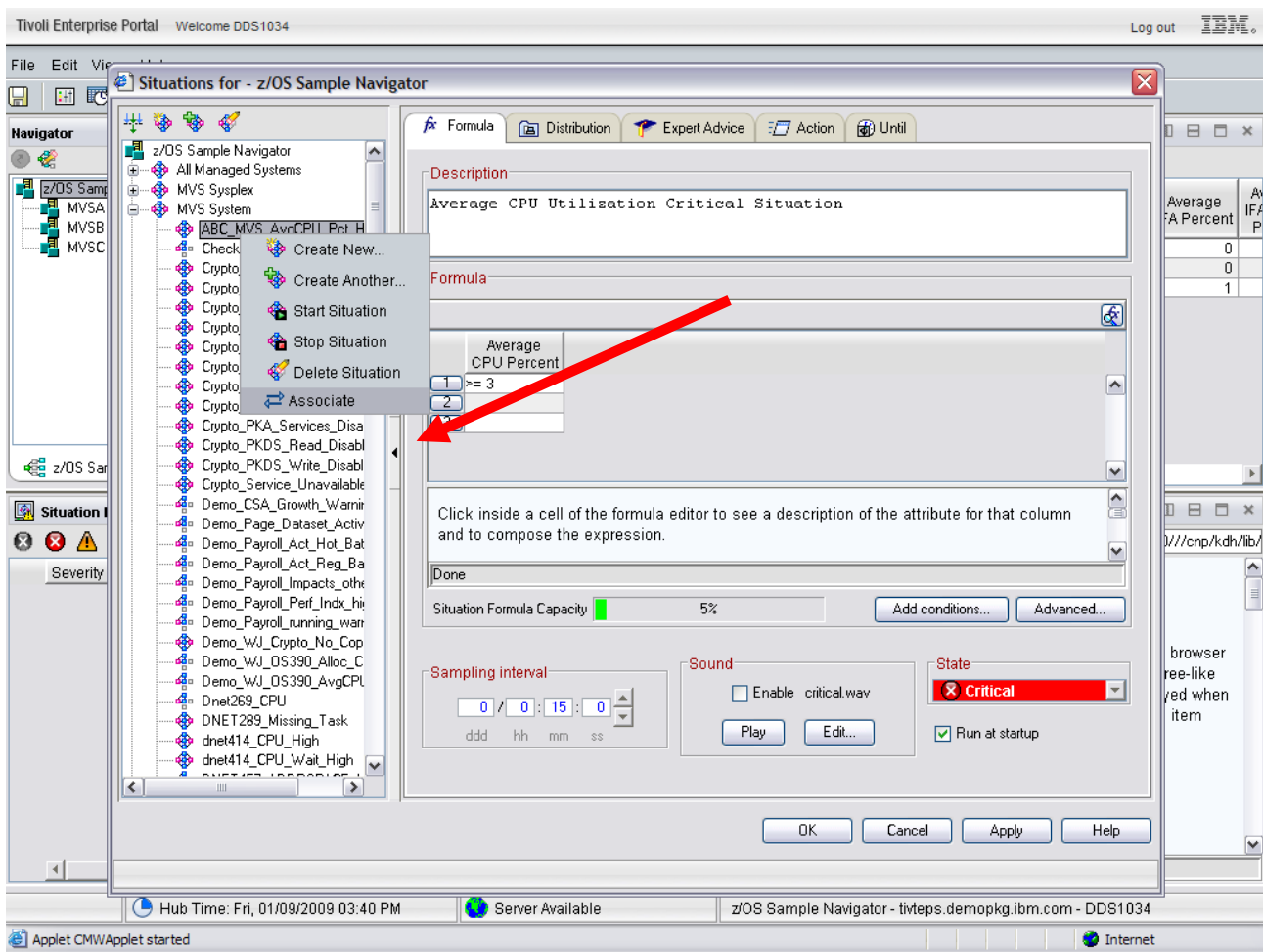


Clicking OK produces the following list of situations.

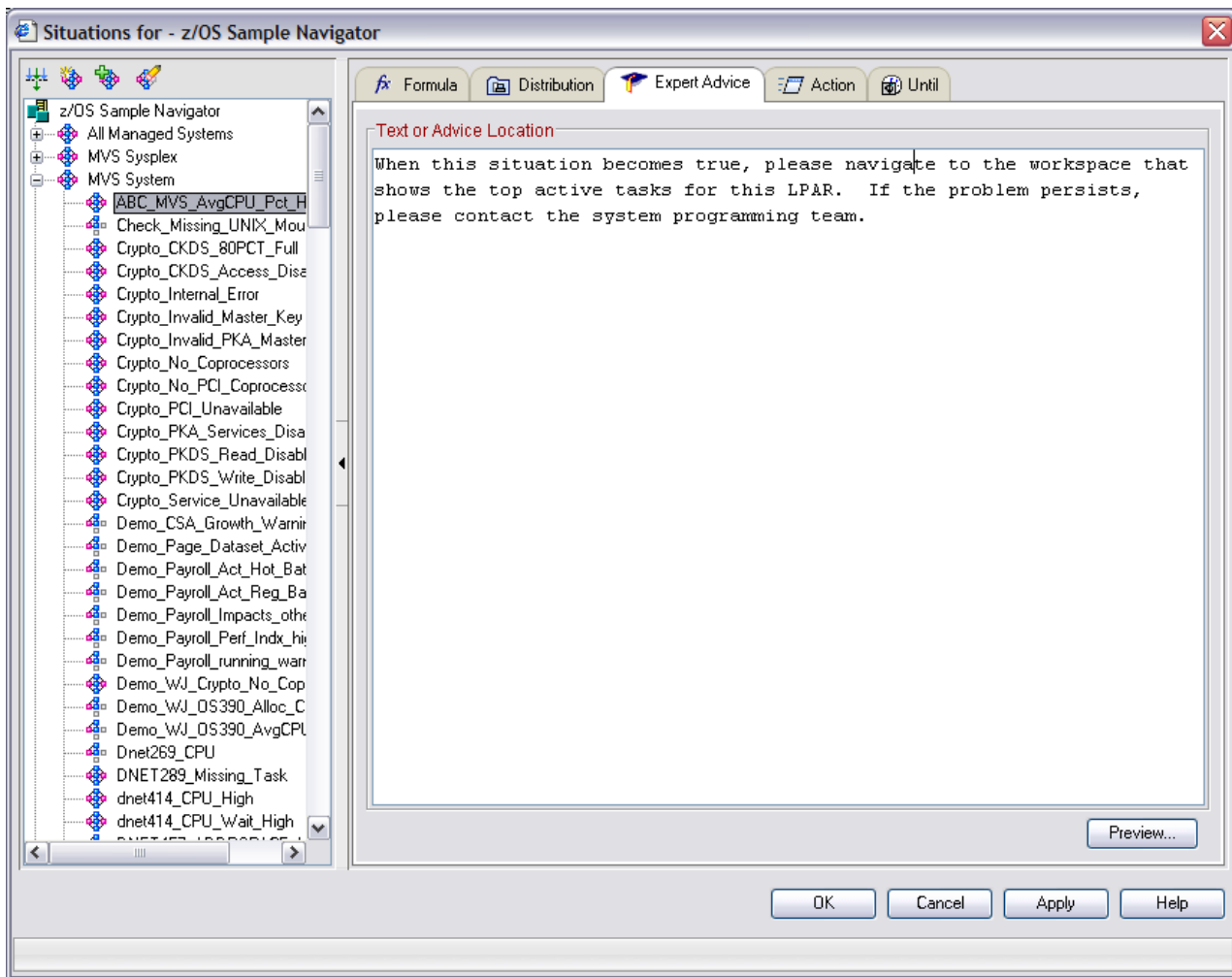


We can see that our custom situation is now in the list and that clicking on it has opened the situation in the situation editor window. Comparing this situation editor window with the one from the last example also shows us that we now have the ability to assign a state or a severity to the event that will be opened when this situation is true. We can now assign a state because we have opened the situation editor from a navigator, rather than from the TEP tool bar.

To associate this situation with the navigator item, we can now right-click the situation name and select "Associate".



The simplest way to pass instructions to the system monitoring group regarding what to do when this problem occurs may be to include simple text Expert Advice in the situation definition. In the next screens, we will add some text expert advice. First, we will select the Expert Advice tab in the situation editor and then simply type in instructions regarding what should be done.



After adding our expert advice, we can click OK to save the situation and then check our workspace to see if an event has opened.

IBM z Systems IBM Monitoring – OMEGAMON Best Practices & Performance Guidelines V2

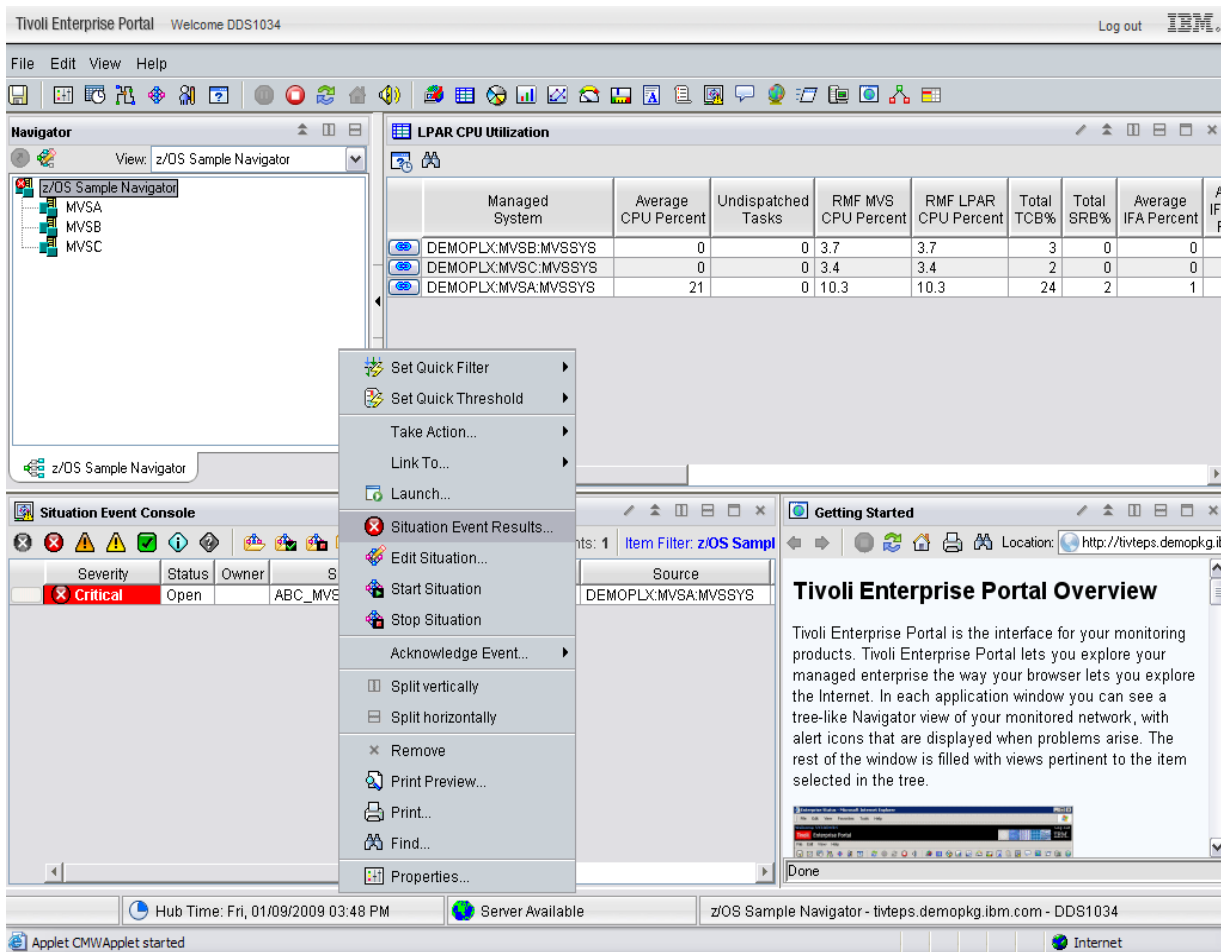
The screenshot displays the Tivoli Enterprise Portal interface. At the top, it says 'Tivoli Enterprise Portal Welcome DDS1034' and 'Log out IBM'. Below the menu bar, there are several panes:

- Navigator:** Shows a tree view of 'z/OS Sample Navigator' with sub-items 'MVSA', 'MVS B', and 'MVS C'. A red arrow points to the 'z/OS Sample Navigator' label at the bottom of this pane.
- LPAR CPU Utilization:** A table showing CPU usage for different managed systems. A red arrow points to the table.
- Situation Event Console:** Shows a table of events. One event is highlighted as 'Critical' with status 'Open'. The event name is 'ABC_MVS_AvgCPU_Pct_High' and the source is 'DEMOPLX:MVSA:MVSSYS'.
- Getting Started:** A pane titled 'Tivoli Enterprise Portal Overview' providing introductory text about the portal. A red arrow points to this pane.

At the bottom of the interface, there is a status bar showing 'Hub Time: Fri, 01/09/2009 03:46 PM', 'Server Available', and 'z/OS Sample Navigator - tivtpeps.demopkg.ibm.com - DDS1034'.

Excellent!! CPU usage of 21% has caused an event to open for MVSA as we can see in our Situation Event Console. To find out exactly why this event opened, we can right-click the event and select Situation Event Results . . . This is shown in the next screen.

IBM z Systems IBM Monitoring – OMEGAMON Best Practices & Performance Guidelines V2



The situation event results workspace opens and a new navigator item is inserted into the navigator with the same name as the situation that has become true. This workspace can be used to discover exactly why the situation became true and what we should do about it.

IBM z Systems IBM Monitoring – OMEGAMON Best Practices & Performance Guidelines V2

Please note that this workspace shows Initial Situation Values, the value of the Average CPU Percent attribute at the time the event was opened, as well as Current Situation Values, the value of the Average CPU Percent attribute at the time the situation event results workspace was opened or last refreshed. A refresh will cause a more current value for Average CPU Percent to appear in the Current Situation Values section, but the value in the Initial Situation Values section will remain constant as long as this event remains open. By hovering over the highlighted attribute, we can see that the situation is true because the value of Average CPU Percent is currently greater than 3%, which we coded in the situation for testing.

We also see sections of the workspace that show the text we entered in the Expert Advice tab of the situation, as well as a Take Action view that can be used to provide some predefined actions. For more information about Take Action, please refer to the [IBM Monitoring User's Guide](#).

The screenshot displays the Tivoli Enterprise Portal interface. At the top, it shows 'Tivoli Enterprise Portal Welcome DDS1034' and a 'Log out' button. Below the menu bar, there are several toolbars. The main workspace is divided into several panes:

- Navigator:** Shows a tree view of the z/OS Sample Navigator with nodes for MVS, MVSA, MVSB, and MVSC. A red arrow points to the 'ABC_MVS_AvgCPU_Pct_High' node.
- Initial Situation Values:** A table showing system metrics. A red arrow points to the 'Average CPU Percent' column, which has a value of 7. Another red arrow points to the 'CPU Flag' column, which has a value of 'RMF'. A yellow tooltip box is visible over the table with the text 'Average CPU Percent >= 3'.
- Current Situation Values:** A table showing updated system metrics. A red arrow points to the 'Average CPU Percent' column, which has a value of 5. Another red arrow points to the 'CPU Flag' column, which has a value of 'RMF'.
- Command View:** A 'Take Action' dialog box with fields for 'Action Name' (set to '<Select Action>') and 'Command'. There is an 'Arguments...' button and a 'Run' button at the bottom.
- Expert Advice:** A text area containing the message: 'When this situation becomes true, please navigate to the workspace that shows the top active tasks for this LPAR. If the problem persists, please contact the system programming team.'

At the bottom of the interface, there is a status bar showing 'Hub Time: Fri, 01/09/2009 03:58 PM', 'Server Available', and the URL 'ABC_MVS_AvgCPU_Pct_High - tivtps.demopkg.ibm.com - DDS1034'. There is also a small 'Applet CMWApplet started' message and an 'Internet' icon.

The system monitoring group can now take the action recommended in the expert advice section by navigating up one level in the navigator to the z/OS Sample Navigator workspace and selecting the Active Tasks link that we created for the MVSA LPAR. This will provide additional information about why CPU utilization is high.

IBM z Systems IBM Monitoring – OMEGAMON Best Practices & Performance Guidelines V2

LPAR CPU Utilization

Managed System	Average CPU Percent	Undispatched Tasks	RMF MVS CPU Percent	RMF LPAR CPU Percent	Total TCB%	Total SRB%	Average IFA Percent	AV IFA P
DEMOPLX:MVSB:MVSSYS	0	0	3.6	3.7	2	0	0	
DEMOPLX:MVSC:MVSSYS	0	0	3.3	3.3	2	0	0	
DEMOPLX:MVSA:MVSSYS	3	0	8.8	8.8	57	1	1	

Situation Event Console

Severity	Status	Owner	Situation Name	Display Item
Critical	Open		ABC_MVS_AvgCPU_Pct_High	DEM

Tivoli Enterprise Portal Overview

Tivoli Enterprise Portal is the interface for your monitoring products. Tivoli Enterprise Portal lets you explore your managed enterprise the way your browser lets you explore the Internet. In each application window you can see a tree-like Navigator view of your monitored network, with alert icons that are displayed when problems arise. The rest of the window is filled with views pertinent to the item selected in the tree.

CPU Usage

Where CPU Percent is greater than 0

System	CPU Percent
OSNMPP	42.6
NETVIEW	31.3
TOPP	26.9
CXEG02	14.7
CXEG02SST	7.8
WLM	1.3
XCFAS	0.4
RMFGAT	0.4
CXEGD2	0.4
CXEGRHA	0.4

Address Space CPU Utilization

Job Name	Step Name	Proc Step	SvcClass	SvcClass Period	ASID	JESJOBID	CPU Percent	TCB Percent	SRB Percent	IFA Percent	IFA on CP Percent	zIIP Percent	zIIP on CP Percent	Independent Enclave CPU%	Independ Enclave IFA%
OSNMPP	OSNMPP	OSNMPP	OPSD	1	0X00E3	STC11178	42.6	42.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0
NETVIEW	NETVIEW	NETV530	OPSD	1	0X0102	STC11224	31.3	31.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0
TCPPIP	TCPPIP	TCPPIP	OPSHI	1	0X0028	STC10980	26.9	26.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0
CXEG02	CXEG02	O2CI	OPSD	1	0X0134	STC11299	14.7	14.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0
CXEGD2SST	CXEGD2SST	TEMS	OPSD	1	0X0133	STC12169	7.8	7.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0
WLM	WLM	IEFPROC	SYSTEM	1	0X000C		1.3	1.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0
XCFAS	XCFAS	IEFPROC	SYSTEM	1	0X0006		0.4	0.4	0.4	0.0	0.0	0.0	0.0	0.0	0.0
RMFGAT	RMFGAT	IEFPROC	SYSTEM	1	0X00B0	STC11124	0.4	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0
CXEGD2	CXEGD2	D2CUA	OPSD	1	0X0135	STC11297	0.4	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0
CXEGRHA	CXEGRHA	KRHMST00	OPSD	1	0X0151	STC11329	0.4	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0

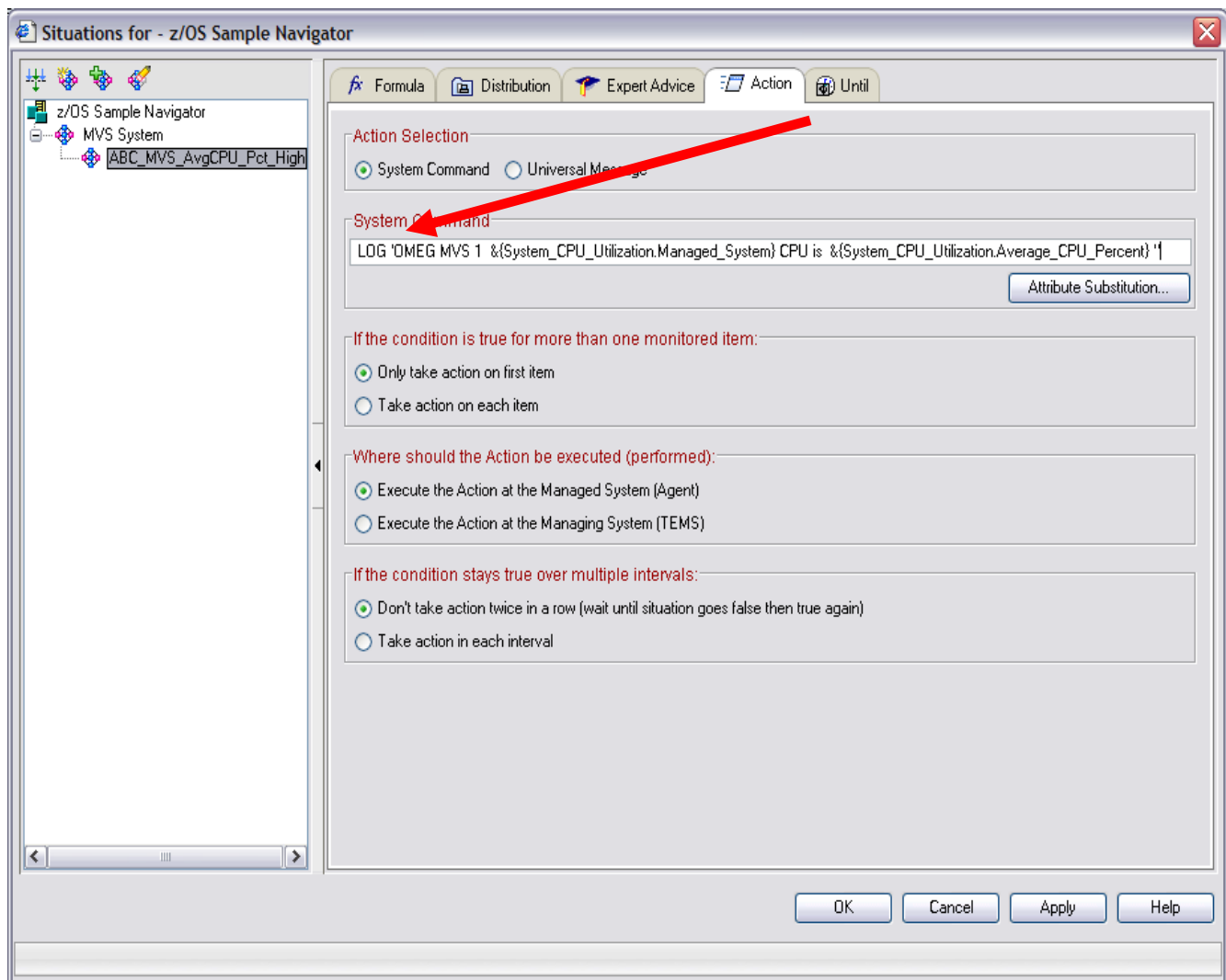
For System MVSA

IBM z Systems **IBM Monitoring – OMEGAMON Best Practices & Performance Guidelines V2**

This concludes the example of manual action for an open event.

An example of how automated action might be taken when a situation becomes true is shown in the following screens.

This time we will modify our situation with the situation editor to perform an automated Take Action. As you can see, the Action tab of the situation editor provides several choices.



The most commonly used is to send one single system command out to the operating system where the agent, or optionally the TEMS to which the agent connects, is running. In the case of OMEGAMON products running on z/OS, it is very common that the agent and the TEMS it reports to are running in the same Z/OS LPAR.

The single command may be a valid Z/OS operator command. D A,L would be one example.

What may be more useful, however, is to provide a text string that can be intercepted by an automated automations package. This text string can be used to drive more sophisticated forms of automation when an event is opened. In the example above, the command sent to Z/OS is LOG. The information that is written to the log includes the word OMEG which indicates an OMEGAMON message, the word Z/OS that tells us this is from the Z/OS monitoring agent, the number 1 which indicates a critical severity, and optional text. In this case, the optional text gives us the name of the managed system (DEMOPLX:MVSA:MVSSYS) and the current value of the Percent CPU Busy attribute.

Depending on the release level and platform of the Hub TEMS in the ITM architecture, it may also be possible to automatically forward ITM OMEGAMON events to enterprise console software products, such as Omnibus or TEC, and managed those events from the enterprise console tool. This is another example of how events may result in automated action.

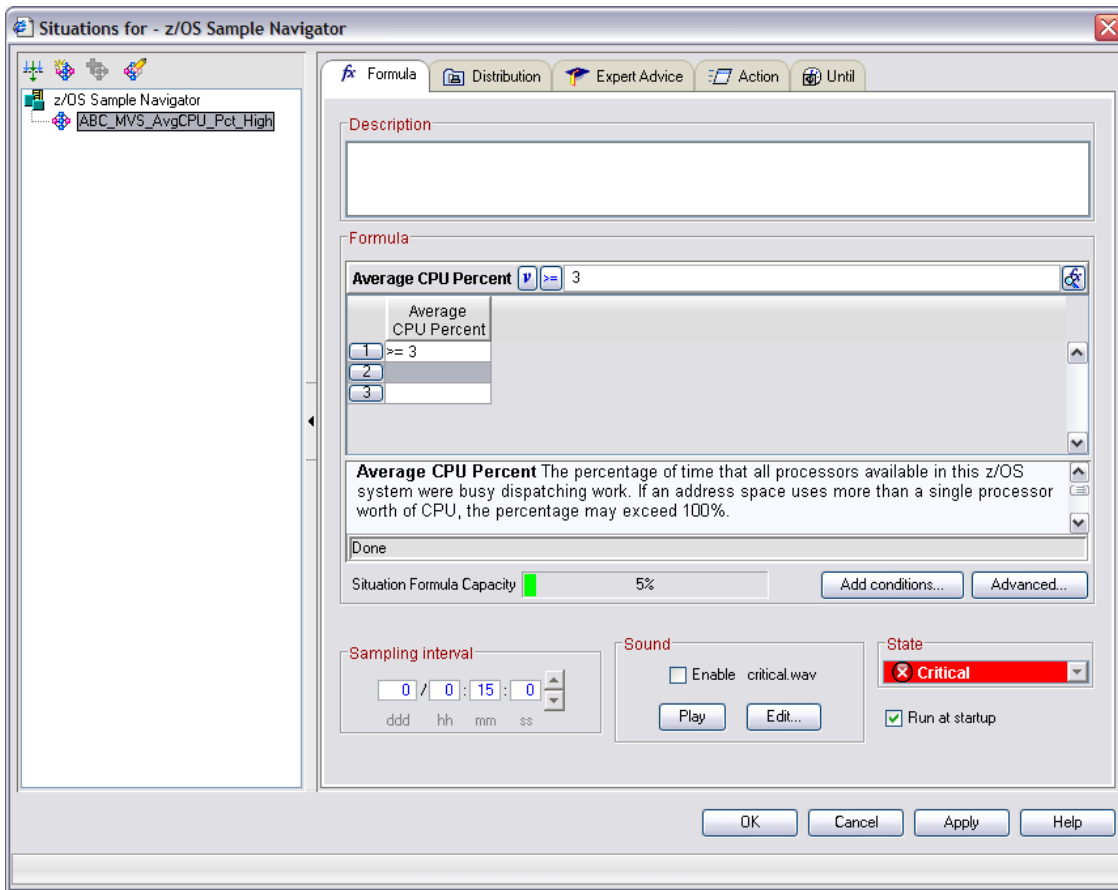
This concludes the discussion of how events should always be associated with some type of action.

Situation advanced functions, such as Persistence and Display Item, can be used to make events more meaningful and reduce the chances of false positive alerts.

Using the custom situation, we previously created, ABC_MVS_AvgCPU_Pct_High, we can now demonstrate how the Persistence feature will provide more meaningful alerts and will likely reduce the possibility of receiving false positive alerts.

When we created the ABC_MVS_AvgCPU_Pct_High situation, we specified checking at 15 minute intervals for system CPU utilization of greater than 3%. (In a real production environment, of course, this threshold value would be 90% or 95% or perhaps even higher, depending on your unique monitoring requirements. But in our laboratory environment, we have set the threshold artificially low to ensure that we will receive alerts.)

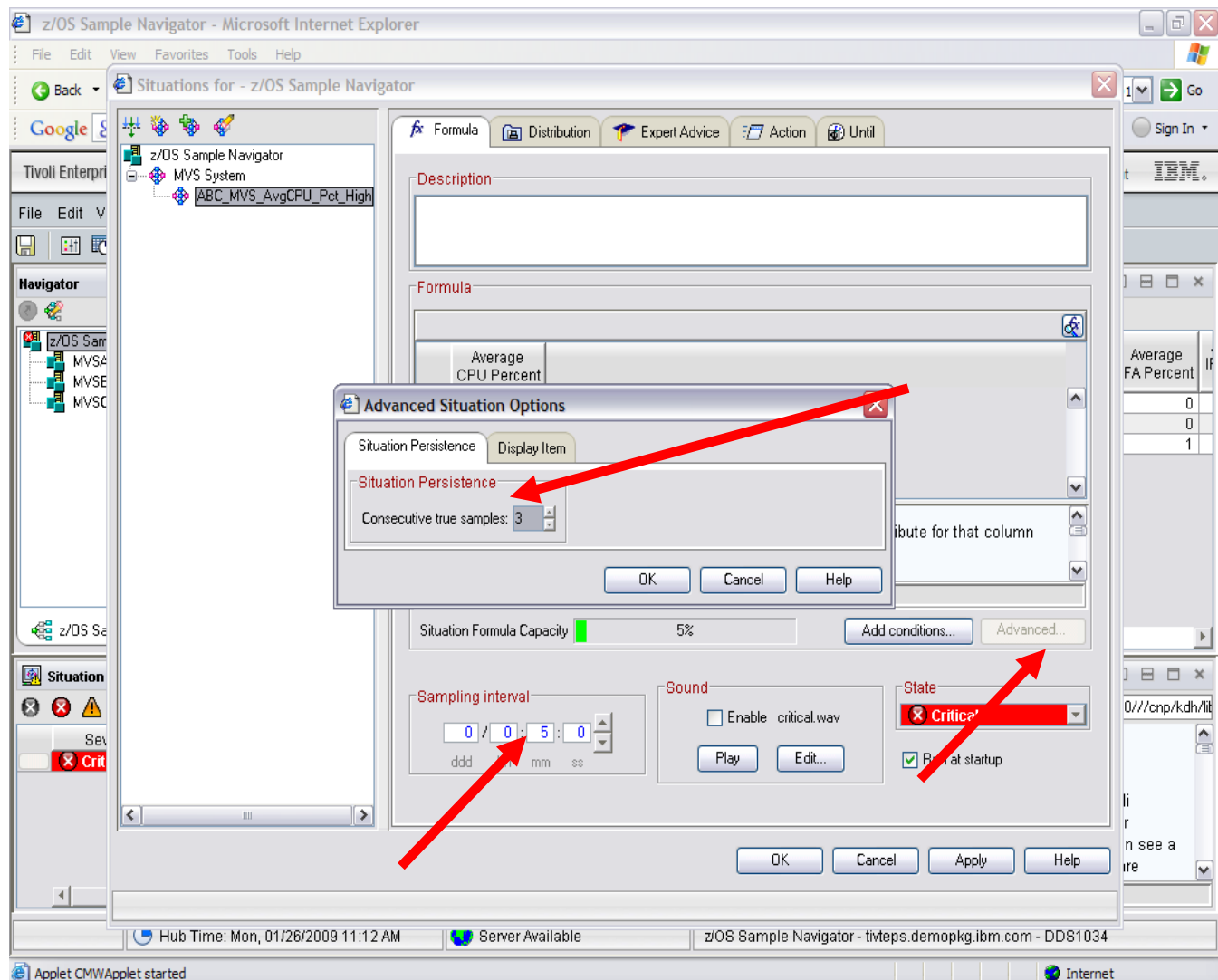
When the situation is evaluated at each 15-minute interval, the current value of the Average CPU Percent attribute is compared to the threshold, in this case 3%.



IBM z Systems IBM Monitoring – OMEGAMON Best Practices & Performance Guidelines V2

This type of situation will, therefore, sometimes be evaluated as true even on a system with a fairly light processing load because the situation will be checked during a workload peak value, rather than on average utilization over time. In other words, you may get alerts even when there is not a problem – a false positive.

One way to deal with this is to use the Advanced. . button to add persistence checking to the situation. We may also wish to decrease the interval at which the situation will be evaluated so that we are sure to get timely notification of potential problems. The follow screen shows the addition of persistence to our situation.



One of the red arrows highlights the Advanced. . button. Another shows that we have used the Situation Persistence tab to set the value of Consecutive true samples to 3, rather than the default of one. This means that the situation will need to be evaluated as true three times in a row before a situation event is opened.

IBM z Systems IBM Monitoring – OMEGAMON Best Practices & Performance Guidelines V2

If we were to leave the Sampling Interval set to 15 minutes, then it would take this situation up to 45 minutes to detect that there was a problem with high CPU. Therefore, we have also changed the Sampling Interval to 5 minutes to ensure that we will receive an alert within 15 minutes of the start of the problem.

In our next example, we will show how the addition of another advanced feature, Display Item, can provide more meaningful alerts. In this example, we have modified our custom workspace to show WLM information for the Sysplex. Now we wish to write a situation that will alert us if any service class is experiencing a high-Performance Index. First, we right-click our z/OS Sample Navigator item and select Situations.

The screenshot displays the Tivoli Enterprise Portal interface. A red arrow points to the 'Situations...' option in the context menu of the 'z/OS Sample Navigator' in the Navigator pane. Another red arrow points to the 'Sysplex WLM Performance' table in the main workspace.

Service Class	Performance Index	Goal	Goal Importance	Period	Actual Host	Worst Performance Index	Transa Rat
UNIX	40.00	Velocity(+I/O) > 40	Medium	2	0	40.00	
DDFDEF	4.10	Pct Resp 80 % < 2.5 s	Medium	1	10250	4.10	
UNIX	1.20	Velocity(+I/O) > 60	High	1	50	1.20	
UNIX	0.75	Velocity(+I/O) > 30	Low	3	40	0.75	
OPSHI	0.73	Velocity(+I/O) > 70	Highest	1	95	1.06	
OPSDEF	0.65	Velocity(+I/O) > 60	High	1	91	0.65	
TSO	0.57	Velocity(+I/O) > 50	Medium	2	87	0.57	
CICSDEF	0.50	Pct Resp 85 % < 1.0 s	High	1	500	0.50	
TSO	0.50	Pct Resp 90 % < 1.0 s	High	1	500	0.50	
BATMED	0.00	Velocity(+I/O) > 20	Low	1	0	0.00	
CBHI	0.00	Avg Resp < 500.0 ms	Highest	1	0	0.00	
OPSLO	0.00	Velocity(+I/O) > 20	Medium	1	0	0.00	
SYSSTC	0.00	Sys Goal	Unavailable	1	0	0.00	
SYSTEM	0.00	Sys Goal	Unavailable	1	0	0.00	

Severity	Status	Owner	Situation Name	Display Item
Critical	Open		ABC_MVS_AvgCPU_Pct_High	DEM

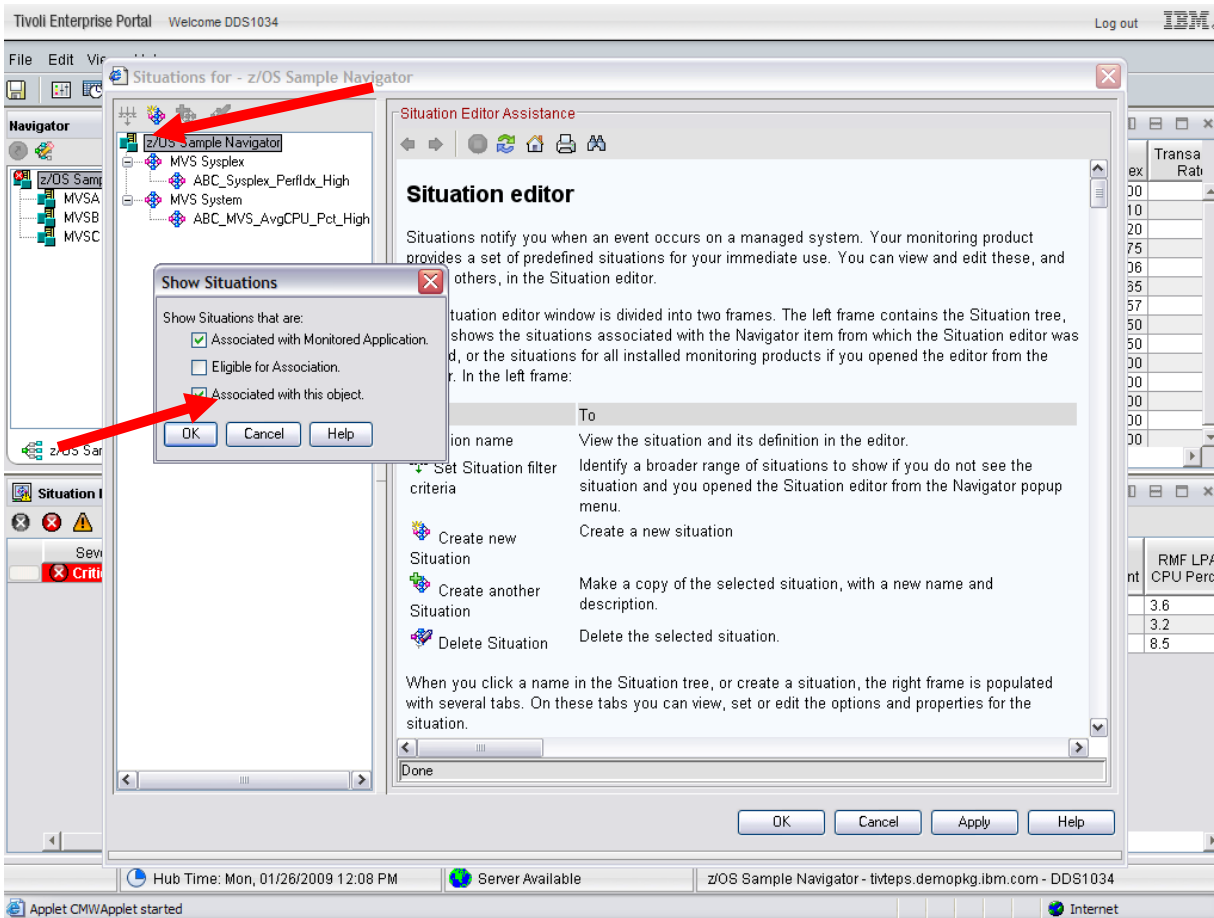
Managed System	Average CPU Percent	Undispatched Tasks	RMF MVS CPU Percent	RMF LPAR CPU Percent
DEMOPLX:MVSB:MVSSYS	0	0	3.6	3.6
DEMOPLX:MVSC:MVSSYS	0	0	3.2	3.2
DEMOPLX:MVSA:MVSSYS	5	0	8.5	8.5

Please also note the new View that has been added to show the current WLM performance metrics for our Sysplex.

IBM z Systems IBM Monitoring – OMEGAMON Best Practices & Performance Guidelines V2

To create our new situation, we will select a product-provided situation that does something similar to what we wish to do and save our own custom copy of that.

The filter tool in the Situation Editor allows us to examine all Sysplex situations to find one that is close to what we wish to do.



On the following screen, you can see that Sysplex_Workloads_PerfIdx_Crit looks a lot like what we wish to monitor. We can use the Create Another . . . icon as highlighted and save our own copy of that situation, named ABC_Sysplex_PerfIdx_High, and make modifications to our new custom situation.

Situations for - z/OS Sample Navigator

Description
Service Class Performance Index Critical

Formula
Performance Index > 1.50

Click inside a cell of the formula editor to see a description of the attribute for that column and to compose the expression.

Situation Formula Capacity 6%

Sampling interval: 0 / 0 : 5 : 0

Sound: Enable critical.wav

State: **Critical**

Run at startup

Create Situation

Name: ABC_Sysplex_PerfIdx_High

Description:

Monitored Application: MVS Sysplex

Correlate Situations across Managed Systems

Situation name:

- 1) Must be 31 characters or less,
- 2) Must start with an alphabetic character (a-z, A-Z),
- 3) May contain any alphabetic, numeric (0-9) or underscore (_) character,
- 4) Must end with an alphabetic or numeric character.

Click inside a cell of the formula editor to see a description of the attribute for that column and to compose the expression.

Situation Formula Capacity 6%

Sampling interval: 0 / 0 : 5 : 0

Sound: Enable critical.wav

State: **Critical**

Run at startup

IBM z Systems IBM Monitoring – OMEGAMON Best Practices & Performance Guidelines V2

To make sure that we get an open situation event in the lab environment, we set our threshold for Performance Index to 0.5, rather than the product-provided value of 1.5. Examining the WLM view in our workspace below tells us that we may get as many as 7 service classes that will exceed this threshold, depending the value of the attributes when this situation is evaluated.

After saving the new situation we see that the situation has been evaluated as true and a new situation event has been opened.

The screenshot displays the Tivoli Enterprise Portal interface. The top section shows the 'Sysplex WLM Performance' table with columns for Service Class, Performance Index, Goal, Goal Importance, Period, Actual Host, Worst Performance Index, and Transa Rat. Below this, the 'Situation Event Console' shows a table of events with columns for Severity, Status, Owner, Situation Name, and Display Item. A red arrow points to the 'Critical' status of the 'ABC_MVS_AvgCPU_Pct_High' event. To the right, the 'LPAR CPU Utilization' table shows columns for Managed System, Average CPU Percent, Undispatched Tasks, RMF MVS CPU Percent, and RMF LPAR CPU Percent.

Service Class	Performance Index	Goal	Goal Importance	Period	Actual Host	Worst Performance Index	Transa Rat
UNIX	40.00	Velocity(+I/O) > 40	Medium	2	0	40.00	
DDFDEF	4.10	Pct Resp 80 % < 2.5 s	Medium	1	10250	4.10	
UNIX	1.20	Velocity(+I/O) > 60	High	1	50	1.20	
UNIX	0.75	Velocity(+I/O) > 30	Low	3	40	0.75	
OPSHI	0.73	Velocity(+I/O) > 70	Highest	1	95	1.06	
OPSDEF	0.65	Velocity(+I/O) > 60	High	1	91	0.65	
TSO	0.57	Velocity(+I/O) > 50	Medium	2	87	0.57	
CICSDEF	0.50	Pct Resp 85 % < 1.0 s	High	1	500	0.50	
TSO	0.50	Pct Resp 90 % < 1.0 s	High	1	500	0.50	
BATMED	0.00	Velocity(+I/O) > 20	Low	1	0	0.00	
CBHI	0.00	Avg Resp < 500.0 ms	Highest	1	0	0.00	
OPSL0	0.00	Velocity(+I/O) > 20	Medium	1	0	0.00	
SYSSTC	0.00	Sys Goal	Unavailable	1	0	0.00	
SYSTEM	0.00	Sys Goal	Unavailable	1	0	0.00	

Severity	Status	Owner	Situation Name	Display Item
Critical	Open		ABC_Sysplex_PerfIdx_High	DEM
Critical	Open		ABC_MVS_AvgCPU_Pct_High	DEM

Managed System	Average CPU Percent	Undispatched Tasks	RMF MVS CPU Percent	RMF LPAR CPU Percent
DEMOPLX:MVSB:MVSSYS	0	0	3.6	3.6
DEMOPLX:MVSC:MVSSYS	0	0	3.2	3.2
DEMOPLX:MVSA:MVSSYS	5	0	8.5	8.5

Now we can right-click that new situation event and select Situation Event Results . . . to navigate to the situation event results workspace. This will allow us to determine why we think we have a problem on our system.

IBM z Systems IBM Monitoring – OMEGAMON Best Practices & Performance Guidelines V2

The screenshot displays the IBM Monitoring console interface. On the left is a 'Navigator' pane showing a tree view of 'z/OS Sample Navigator' with sub-items: ABC_Sysplex_PerflDx_High, MVSA, MVSB, and MVSC. The main area is divided into several panes:

- Initial Situation Values:** A table showing performance metrics for five service classes. The first row (DDFDEF) has a Performance Index of 4.10. The last row (UNIX) has a Performance Index of 1.11.
- Current Situation Values:** A table showing updated performance metrics. The first row (DDFDEF) has a Performance Index of 4.10. The last row (UNIX) has a Performance Index of 0.88.
- Command View:** A 'Take Action' dialog with fields for 'Name' (set to '<Select Action>') and 'Command', and a 'Destination Systems' list.
- Expert Advice:** A pane titled 'Expert Advice' showing a situation description: 'Sysplex_Workloads_PerflDx_Crit' with a 'Service Class Performance Index Critical' status and 'Suggested Actions'.

At the bottom of the console, there is a status bar showing 'Hub Time: Mon, 01/26/2009 12:16 PM', 'Server Available', and the current situation name: 'ABC_Sysplex_PerflDx_High - tivtpeps.demopkg.ibm.com - DDS1034'.

In the situation event results workspace, we see that the Initial Situation Values view shows us that 5 service classes exceeded the threshold when the situation was evaluated and the event was first opened.

However, the view for Current Situation Values shows us that when our situation event results workspace was opened, or last updated, there were several differences in the service classes that are represented as problems in this event. UNIX period one workload is no longer in our list of current situation values, but TSO period 2 has been added since the situation was first evaluated.

The significance of this type of behavior is that once a situation event has opened, it will remain open as long as there is even one service class with a performance index greater than 0.5. It may be a different set of service classes each time the situation is evaluated, but the event will remain open until no service class has a performance index greater than 0.5.

This makes it more difficult to determine the real cause of our problem without additional effort, and it may also make it more difficult to define automated actions that should be taken as a result of this situation event.

In the next screen, we will edit our situation to add a display item and see how the results differ.

We have selected the situation we wish to change, clicked the Advanced. . button, selected the Display Item tab in the Advanced Situation Options dialog box, and selected the Service Class attribute from the pull-down list. Then we can click OK, then OK again to save the situation and see how this effects our situation event.

Tivoli Enterprise Portal Welcome DDS1034

Situations for - z/OS Sample Navigator

Formula

Description: Service Class Performance Index Critical

Formula

Advanced Situation Options

Display Item

Group: Sysplex_WLM_Service_Class_Period

Item: Service Class

Sampling interval: 0 / 0 : 5 : 0

Sound: Enable < none selected >

State: **Critical**

Run at startup:

Transa	Rab
10	
20	
30	
72	
36	
53	
50	
50	
22	
00	
00	
00	
00	
00	
00	

RMF LPAR	CPU Perce
3.6	
8.1	

Hub Time: Mon, 01/26/2009 12:28 PM Server Available z/OS Sample Navigator - tivtpeps.demopkg.ibm.com - DDS1034

Applet CMWApplet started Internet

IBM z Systems IBM Monitoring – OMEGAMON Best Practices & Performance Guidelines V2

We can see that the addition of Display Item has caused a unique situation event to open for each of the service classes that exceeds our threshold of 0.5. We can also now see the name of the service class that caused each situation event to open, right in our situation event console. Since we now have unique events for each condition that we consider to be a problem, we can manage each event individually, either manually or with automation. We can also be confident that each unique event will open when a problem occurs and close when the problem is resolved, rather than just one single event that persists until all potential problems are cleared.

The screenshot displays the Tivoli Enterprise Portal interface. The top navigation bar includes 'Tivoli Enterprise Portal', 'Welcome DDS1034', and 'Log out IBM'. Below the menu, there are several toolbars and panels:

- Navigator:** Shows a tree view of 'z/OS Sample Navigator' with sub-items MVSA, MYSB, and MVSC.
- Sysplex WLM Performance:** A table listing performance metrics for various service classes.

Service Class	Performance Index	Goal	Goal Importance	Period	Actual Host	Worst Performance Index	Transa Rat
DDFDEF	4.10	Pct Resp 80 % < 2.5 s	Medium	1	10250	4.10	
UNIX	1.20	Velocity(+I/O) > 30	Low	3	25	1.20	
UNIX	0.80	Velocity(+I/O) > 40	Medium	2	50	0.80	
OPSHI	0.72	Velocity(+I/O) > 70	Highest	1	97	0.72	
OPSDEF	0.65	Velocity(+I/O) > 60	High	1	91	0.66	
TSO	0.53	Velocity(+I/O) > 50	Medium	2	93	0.53	
CICSDEF	0.50	Pct Resp 85 % < 1.0 s	High	1	500	0.50	
TSO	0.50	Pct Resp 90 % < 1.0 s	High	1	500	0.50	
BATMED	0.22	Velocity(+I/O) > 20	Low	1	88	0.22	
CBHI	0.00	Avg Resp < 500.0 ms	Highest	1	0	0.00	
OPSL0	0.00	Velocity(+I/O) > 20	Medium	1	0	0.00	
SYSSTC	0.00	Sys Goal	Unavailable	1	0	0.00	
SYSTEM	0.00	Sys Goal	Unavailable	1	0	0.00	
UNIX	0.00	Velocity(+I/O) > 60	High	1	0	0.00	
- Situation Event Console:** Shows a list of critical events.

Severity	Status	Owner	Situation Name	Display Item
Critical	Open		ABC_Sysplex_PerfIdx_High	UNIX
Critical	Open		ABC_Sysplex_PerfIdx_High	TSO
Critical	Open		ABC_Sysplex_PerfIdx_High	OPSHI
Critical	Open		ABC_Sysplex_PerfIdx_High	OPSDEF
Critical	Open		ABC_Sysplex_PerfIdx_High	DDFDEF
Critical	Open		ABC_MVS_AvgCPU_Pct_High	
- LPAR CPU Utilization:** A table showing CPU usage for managed systems.

Managed System	Average CPU Percent	Undispatched Tasks	RMF MVS CPU Percent	RMF LPAR CPU Perce
DEMOPLX:MVSA:MVSSYS	0	0	3.6	3.6
DEMOPLX:MVSC:MVSSYS	0	0	3.2	3.3
DEMOPLX:MVSA:MVSSYS	8	0	8.1	8.1

Red arrows in the screenshot point from the 'Display Item' column in the Situation Event Console to the corresponding rows in the LPAR CPU Utilization table.

This concludes the discussion of situation advanced functions.

19. Architecture Planning and deployment strategy

19.1 Architecture Components - Constraints

- 1 Hub TEMS, a Remote TEMS for each LPAR, between 1500-3000 managed systems representing approximately 900 heartbeat s. No single Remote TEMS should have more than 400 heartbeat agents (not sub-nodes, CICS Regions, QMGRs, etc.) – less than the 500 due to anticipated event traffic and historical data collection.
- The Remote TEMS can be co-located with a device running TEMAs.
- Multiple TEPS might be considered to support the various physical locations and/ or the user count. The TEPS database can be replicated daily from the MASTER TEPS to the SLAVE TEPS.
- Future Consideration: MQSeries end-points not running on the zOS platform can report to either a UNIX or Windows Remote TEMS.
- Any mid-tier agents should report to a mid-tier Remote TEMS. Mid-tier Remote should be limited to 300 agents.

19.2 Deployment Goals

- The final goal of a single HUB TEMS per ENTERPRISE LEVEL should be guiding direction.
- The HUB TEMS should be located on LINUX on Z – using LDAP validation to a z/OS LDAP Server.
- Fail-over/ redundancy is to be provided for the TEMA – TEMS components. During the rollout, configuration of a small environment should be constructed with redundancy configured for the REMOTE TEMS and TEPS components. Several tests/ scenarios should be run to ensure the failover functions as expected to ensure the configuration is proper.
- Limit user functionality and information within the TEPS by product group.

19.3 Deployment Guidelines

- The v6.3 HUB TEMS should be built first using a production ready box.
- A fail-over HUB TEMS should also be constructed early in the process if required.
- Test the failover mechanism thoroughly.
- To ensure all communication variations are accounted for, the initial rollout should be kept to as small and controllable volume as possible. This would include the HUB TEMS, Remote TEMS to cover (DMZ-Internal Firewall Zones), and a small number of agents reporting to each of the Remote TEMS.
- The zOS agents should be upgraded to the v5.x level as they move to the ITM v6.3 TEMS structure.

19.4 Locating and sizing the hub TEMS

Always locate the hub monitoring server in a data center with good network reliability and throughput. Network connection speeds of 100 Megabits per second or higher are typically sufficient.

Except where noted, the server sizing assume that the hub monitoring server is installed on a separate server with no additional IBMi® components. Below are some basic guidelines. For detailed guidelines, see [Hardware and software requirements](#).

Sizing your hub monitoring server hardware

Large and medium-sized environments (>1500 agents)

Remote TEMSs should be used in this environment. The Hub TEMSs should have at least 2 processors and 3 GB of RAM.

Although the steady-state CPU utilization is usually low, the hub monitoring server uses a significant amount of system resources in processing transient workloads, such as startup, agent login, and situation distribution. It is reasonable to run other IBM Monitoring components on the same server such as the portal server, but allocate additional CPUs for the additional components. For more information, see [Memory and disk requirements](#).

Small environments (< 1000 agents)

Remote TEMSs are not typically required in this environment. The system should have at least 2 processors and 4 GB of RAM. It is reasonable to combine components such as the monitoring server and portal server on a dual-processor server. When combining monitoring components on a single server, the different components' process memory requirements should be added together. See [Memory and disk requirements](#) for more information.

For environments with 200 or fewer agents, most of the components can be combined onto a single server. If you are going to use a small monitoring environment, choose a multiprocessor (2 or 4-way) for the monitoring server with at least 4 GB of total system memory. Configure the software and begin monitoring CPU and memory usage during periods of both normal and high volumes of situation events. If CPU or memory usage is constrained, consider deploying a separate server for the TDW, the Warehouse Proxy Agent, and the Summarization and Pruning Agent. For more information see [Memory and disk requirements](#).

19.5 When to use multiple hub monitoring servers

As environments grow in size, you must run multiple hub monitoring servers. The recommended limit of managed systems per hub monitoring server is 20,000.

There are many factors that affect the scalability of a hub environment. The primary factors are network latency, number of situations running at the hub, the number of concurrent active users, and historical data collection.

[http://www-](http://www-01.ibm.com/support/knowledgecenter/SSTFXA_6.3.0.2/com.ibm.itm.doc_6.3fp2/install/sizehardware.htm?lang=en)

[01.ibm.com/support/knowledgecenter/SSTFXA_6.3.0.2/com.ibm.itm.doc_6.3fp2/install/sizehardware.htm?lang=en](http://www-01.ibm.com/support/knowledgecenter/SSTFXA_6.3.0.2/com.ibm.itm.doc_6.3fp2/install/sizehardware.htm?lang=en)

19.6 Locating and sizing the remote TEMS

When determining the location of your remote monitoring server, you must consider the number of connected agents, existence, number, and placement of firewalls, and network bandwidth. With the IBMi® Monitoring V6.3 release, as a general planning guideline, a single remote monitoring server can support up to 1500 managed systems.

The main factor affecting remote monitoring server scalability is situation processing, and the volume and frequency of agent attribute group data that is processed by the monitoring server during situation evaluation. Situation processing affects monitoring server performance, so the CPU and memory usage of the kdsmain process should be monitored. For tuning recommendations for improving situation processing performance, see [Optimizing situations](#).

Although the steady state CPU utilization is usually low, a remote monitoring server uses a significant amount of system resources in processing transient workloads, such as startup, agent login, and situation distribution. Dual processor systems are recommended for remote monitoring servers. To support 1500 managed systems, a server with 4 GB of

memory is typically sufficient. For environments with more than 3000 agents collecting historical data, you can use multiple Warehouse Proxy agents, with the Warehouse Proxy agents located on each remote monitoring server to handle historical data exports from agents connected to the remote monitoring server.

If the remote deployment capability is used, ensure that the remote monitoring server has sufficient network capabilities (100 Megabits per second or higher is typically sufficient). Because the remote deployment packages are large, in some cases hundreds of megabytes, you must ensure that you do not have a network bottleneck.

The network connection speed between the monitoring server and the monitoring agent will affect the elapsed time for remote deployment operations. One way to deal with slow network links is to deploy a remote monitoring server near the monitored servers. By co-locating the remote monitoring server with the agents, the Remote Deploy bundles need to be transferred across the slow link only once. Distribution to the agents can then be done on a high-speed LAN segment.

19.7 Optimizing situations

Situations are conditions that you want to monitor on managed systems. Each situation contains a name, a predicate formula, special attributes for specific situation processing, information about whether the situation is automatically started or not, and the sampling interval. It can also contain a command to execute when the situation is true, and advice to give the client when an alert for the situation is surfaced, and so on.

A situation predicate is a formula that uses attributes, functions, and other situations as operands along with threshold values to describe a filtering criterion. The situation formula is made up of one or more logical expressions.

Each expression takes the form:

- [Attribute name] / [logical operator] / [value]
- For example: PROCESS_ID == 0

The situation predicates are similar to a WHERE clause in SQL. In IBM® Monitoring, predicates are processed sequentially, and perform better if you put the most restrictive condition as the first predicate. You can use the situation editor to create/edit situations by choosing attributes, logical operator, value, and sampling interval, and so on. Situations are assigned to run on behalf of managed systems or lists of managed systems.

When a situation is running on a monitoring agent, the agent collects the current values of the attributes specified in the formula, and tests the values against a threshold. When the condition is met, that is the threshold is exceeded or a value is matched, the agent passes the collected data back to the monitoring server to which it is connected, and an event is generated.

However, some types of situations cannot be tested at the agent level:

- Situations involving group functions, such as MIN, MAX, AVG, SUM, COUNT
- Embedded situations
- Correlated situations

For these types of situations, the agent returns all rows back to the monitoring server to which it is connected, and the server performs the testing. In large-scale environments, especially if the sampling interval for the situation is short, evaluation for such situations dramatically increases the workload and memory usage for the monitoring server.

In general, there is limited situation processing at the hub monitoring server if there are no agents directly connected to the hub. For remote monitoring servers, there is a direct correlation to the number of agents, situations, and data rows to the amount of memory required. Therefore, the number of situations and size of the data might be the limiting factor that determines how many agents an individual remote monitoring server can support.

Since the performance of the monitoring server is greatly affected by the volume and frequency of situation state changes, do not run a situation and collect data unless you are interested in the results.

The following recommendations provide guidance for how to write more efficient situations and how to reduce the situation processing requirements:

1. If possible, avoid use of group functions, embedded situations, or correlated situations. Processing demands for such situations are much higher, since all attribute group rows must be sent to the monitoring server for testing, increasing the processing demand and memory usage on the monitoring server.
2. Put the most stringent condition at the beginning of the formula because the conditions are evaluated sequentially, left to right.

Consider a situation that has the first condition test on real storage use, the result set may contain multiple rows; then the second condition tests whether a given process name is among the returned rows. It would be more efficient to first test on process name (the result will be one row), followed by the test on the storage usage, just on the single row.

Consider the following in creating the condition tests:

- a. Numeric attributes are processed more quickly than text attributes.
- b. 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.
3. Use longer sampling intervals where possible, especially for situations that are distributed to many agents.
4. Minimize the number of situations on attribute groups that can return many rows, such as process or disk attribute groups.
5. Put mission critical systems in a separated managed system group, and distribute the heavy workload situations to them only if necessary.
6. Consider spreading business-critical situations among several different remote TEMSs. For example, a database monitoring situation might be high load, but business-critical. Instead of having all 500 database agents report through the same remote monitoring server, consider configuring the database agents across five remote monitoring servers, which are supporting other less-demanding agents.

19.8 Locating and sizing the TEPS

Install the portal server on a very fast server with at least 2 processors.

The IBM® Monitoring V6.3 release has been tested with 100 concurrent portal clients on 64-bit platforms. See the [TEPS](#) for more details.

For optimal performance, place the portal server in the same data centers as most of the end users. Depending on firewall restrictions the location of the portal server has to be verified. If clients are not allowed to connect to a central segment they are not able to use the portal server.

If you have multiple large data centers you might need to install a second read-only portal server. If you choose to use this option, be aware that only one portal server can be used to make customization changes such as editing workspaces and situations. See the section on high availability for details on how to configure multiple portal servers.

19.9 Memory and disk requirements

The following table shows estimated memory and disk storage for IBM Monitoring components on distributed systems. The estimated memory usage for each component does not include the operating system and any concurrently running applications. To estimate the memory requirements for a machine, add the memory requirements for each component that will run on the machine, and add at least 1 GB for the operating system and to allow memory for file system caching.

Add the sizings for individual components to calculate a total for more than one component installed on the same computer. For example, if the hub monitoring server and portal server are installed on the same machine, the base component memory requirement is 1600 MB with an additional 220 MB per 1000 managed systems. For an environment with 2000 managed systems, the combined component memory requirement is $1600 + 2 * 220 = 2040$ MB. The machine should be configured with at least 4 GB RAM (2040 MB + 1 GB for the OS and file system caching + extra memory for dynamic requests and some situation evaluation on the hub monitoring server).

Table 1. Estimated memory and disk storage for IBM Monitoring components on distributed systems

Component	Memory requirements ¹		Disk storage requirements ²
	Base memory requirement	Additional memory requirements	
TEMS	100 MB	120 MB per 1000 managed systems in environment. Extra memory for situations evaluated at monitoring server. ⁶ Extra memory for large dynamic data requests. ⁷	1.5 GB ^{3 5}
Remote TEMS	100 MB	120 MB per 100 managed systems connecting through the server. Extra memory for situations	1.0 GB xLinux ^{3 5} 1.5 GB All other systems ^{3 5}



Table 1. Estimated memory and disk storage for IBM Monitoring components on distributed systems

Component	Memory requirements ¹		Disk storage requirements ²
	Base memory requirement	Additional memory requirements	
		evaluated at monitoring server.	
TEMS for Automation	100 MB	70 MB per 1000 managed systems in environment.	100 MB this does not include all of the additional Jazz™ for Service Management components 8
TEPS	1500 MB	100 MB per 1000 managed systems in environment (for AIX® systems, use 150 MB per 1000 managed systems). 4 Extra 800 MB if the dashboard data provider is enabled. Extra memory for large dynamic data requests. 7	4.0 GB plus an additional 1.2 GB in your computer's temporary directory to install the eWAS server and the Eclipse Help Server 5
Portal client (browser or desktop)	400 MB	Extra 200 MB for large environments.	150 MB
TDW	4 GB depending on database configuration parameters	8 - 16 GB depending on database configuration parameters	See Estimating the required size of your database.
Warehouse Proxy Agent	200 MB		150 MB
Summarization and Pruning Agent	500 MB		150 MB

Table 1. Estimated memory and disk storage for IBM Monitoring components on distributed systems

Component	Memory requirements ¹		Disk storage requirements ²
	Base memory requirement	Additional memory requirements	
IBM Performance Analyzer	200 MB	Add 1.5 GB for large environments.	Depends on TDW configuration for IBM Performance Analyzer attribute groups
Dashboard Application Services Hub server (for use with IBM® Infrastructure Management Dashboards for Servers)	1000 MB		3.0 GB 8

Note:

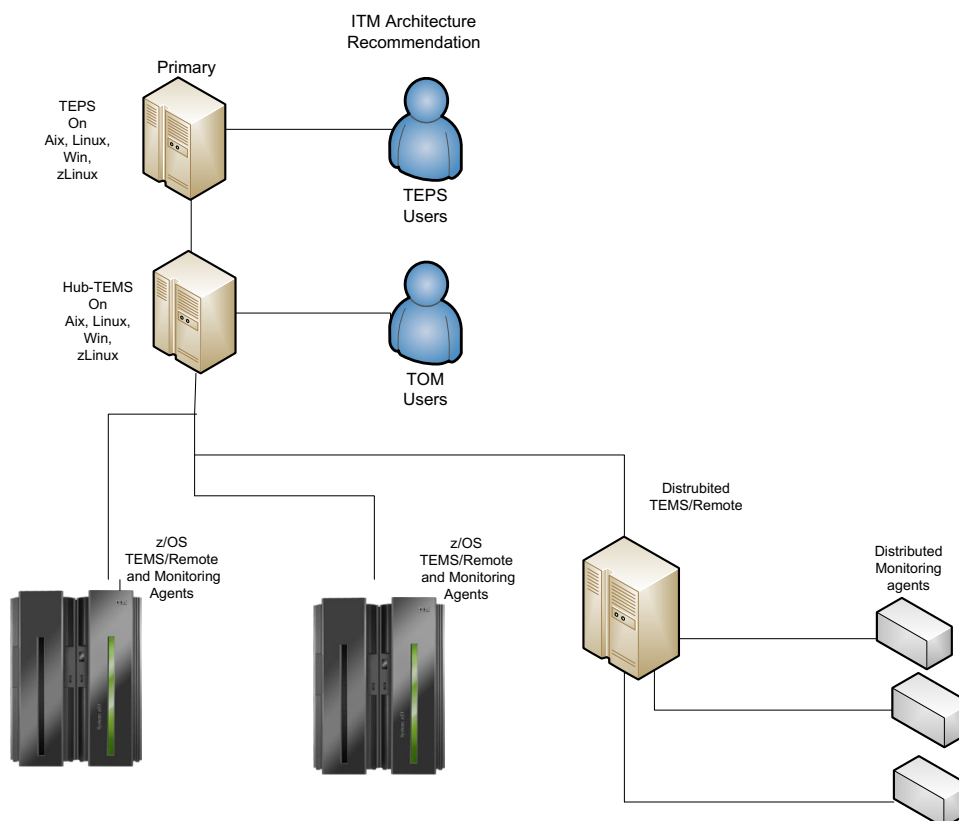
1. The memory and disk sizings shown in this table are the amounts required for the individual component beyond the needs of the operating system and any concurrently running applications. For the total system memory required by small, medium-size, and large environments, see [Sizing your IBM Monitoring hardware](#).
2. The disk storage estimates apply to any size monitoring environment and are considered high estimates. The size of log files affect the amount of storage required.
3. The disk storage requirements for the hub and remote monitoring servers do not include storage for the agent depot, which can require an additional 1 GB or more.
4. The memory requirement for the portal server does not include database processes for the portal server database. DB2® memory usage varies from machine to machine, and can be controlled through various tuning parameters, including bufferpool size. DB2 systems frequently have self-tuning enabled to control the bufferpool size. On systems with a large amount of memory this can result in bufferpools that are much larger than necessary for the TEPS database. A good starting bufferpool size for the TEPS database is 10,000 4K pages (about 40MB). Manual control of the TEPS bufferpool size is recommended to reduce memory usage. You can use the DB2 ALTER command to change the buffer pool size. For example:
 5. Connect TO TEPS:
 6. ALTER BUFFERPOOL IBMDEFAULTBP IMMEDIATE SIZE 10000;
 - CONNECT RESET:
7. IBM Monitoring components require additional disk storage when the self-describing agent feature is enabled. The self-describing agent disk storage usage sizes should be multiplied by the number of products in your environment that are expected to perform self-describing agent installations. This estimate is only applicable when the self-describing agent feature is enabled. The disk storage is used by the directory indicated in the TEMS variable TEMS_MANIFEST_PATH, or the TEPS variable TEPS_MANIFEST_PATH.
 - The hub monitoring server requires an estimated 3 to 5 MB of additional disk storage for each product that is performing self-describing agent installations. WebSphere® and OMEGAMON® agents require more storage, so use the 5 MB per product estimate if you are installing those agents. Otherwise, you can use the 3 MB average size per product.

- The remote monitoring server requires an estimated 2 MB of additional disk storage for each product that is performing self-describing agent installations.
 - The portal server requires an estimated 3 to 6 MB of additional disk storage for each product that is performing self-describing agent installations since it maintains the new and previous version of an agent product's application support. WebSphere and OMEGAMON agents require more storage, so use the 6 MB per product estimate if you are installing those agents. Otherwise, you can use the 3 MB average size per product.
8. Use the following formula to estimate the additional memory required for a situation evaluated at the monitoring server: $2 * \text{number of agents running the situation} * \text{number of row instances evaluated per situation interval} * \text{row size for attribute group}$. The row size for an attribute group is available in the agent user's guide. As an example, assume an environment has 3000 Linux agents, each running a situation that uses the COUNT function to calculate the number of disks reported by the Linux_Disk attribute group. Assume also that the agents report on average of 15 rows for Linux_Disk, and the row size for Linux_Disk is 250 bytes. The additional memory required at the monitoring server for this situation would be: $2 * 3000 * 15 * 250 = 22,500,000$ bytes.
 9. Requests to the monitoring server for a large amount of data can increase the dynamic memory usage significantly. For example, if a user request results in 10,000 rows of Linux_Disk data (250 bytes each) being returned, the additional memory usage would be: $10,000 * 250 = 2,500,000$ bytes. In sizing the memory requirements for the monitoring server, consider increasing the memory estimate by 50% to allow room for dynamic memory requests.
 10. See the Jazz for Service Management hardware and software requirements at http://pic.dhe.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.psc.doc_1.1.0/install/psc_c_install_prereqs.html

19.10 Architecture Guidelines

This section explains the overall architecture of IBM OMEGAMON. The architecture is based on IBM OMEGAMON V5.x and ITM V6.3. In this section, we explain the structure and mechanism of IBM OMEGAMON monitoring and the interrelationship and dependencies of its main components.

IBM OMEGAMON products on z/OS consist of several independent components that make up the overall monitoring environment as shown in the following figure.



As shown in the previous figure these are the components of IBM OMEGAMON:

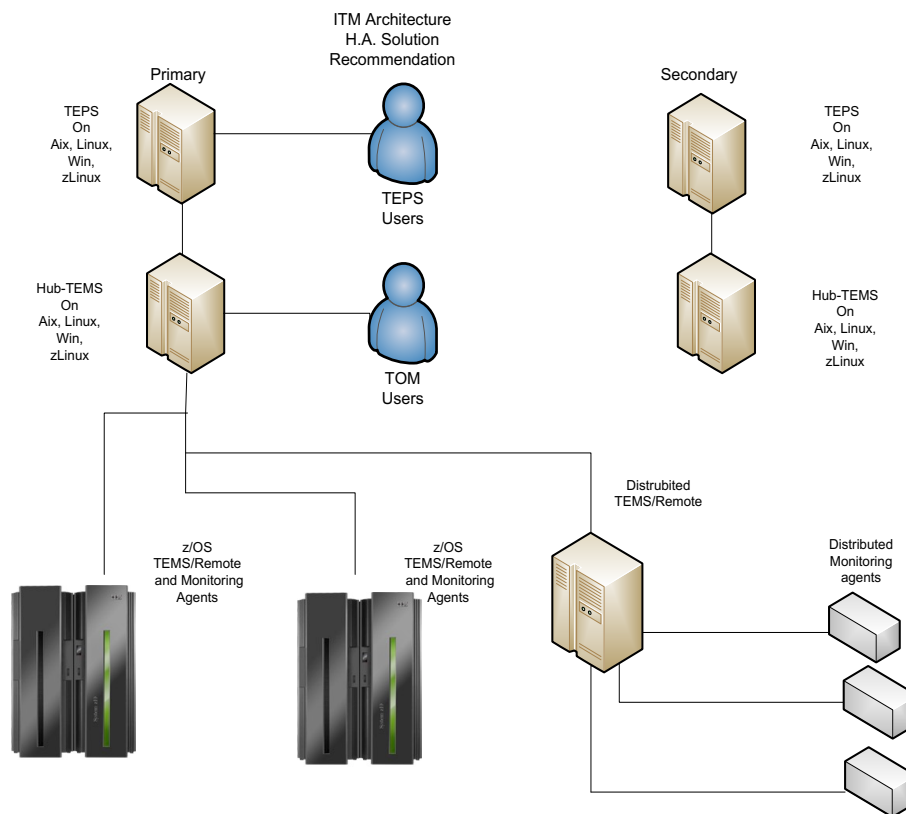
- TEP, a Java based user interface that communicates with TEPS to request and retrieve monitoring data. Depending on how it is installed, you can view the data using a desktop application (desktop mode) or Web browser (browser mode).
- TEPS, a Java application server that enables retrieval, manipulation, 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 TEMS and can be accessed by the TEPclient.
- TEMS has these functions:
 - Consolidates the data collected by the agent and distributes the data either to the TEPS or to the Hub TEMS if it is a remote TEMS.
 - Evaluates column function for situations that cannot be resolved at the TEMA.
 - Sends an alert to the TEPS when specified conditions are met and receives commands from the interface and distributes 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 TEMS Enterprise Information Base (EIB) tables when a monitoring agent is installed.
 - Specifies security settings such as a log on password requirement.
- TEMA 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 TEMS. Actions can be performed locally on TEMA. These agents also respond to and process queries from TEPS. These

IBM z Systems **IBM Monitoring – OMEGAMON Best Practices & Performance Guidelines V2**

agents can issue commands to the monitored system or application. This document discusses the monitoring agent specifically for the z/OS environment.

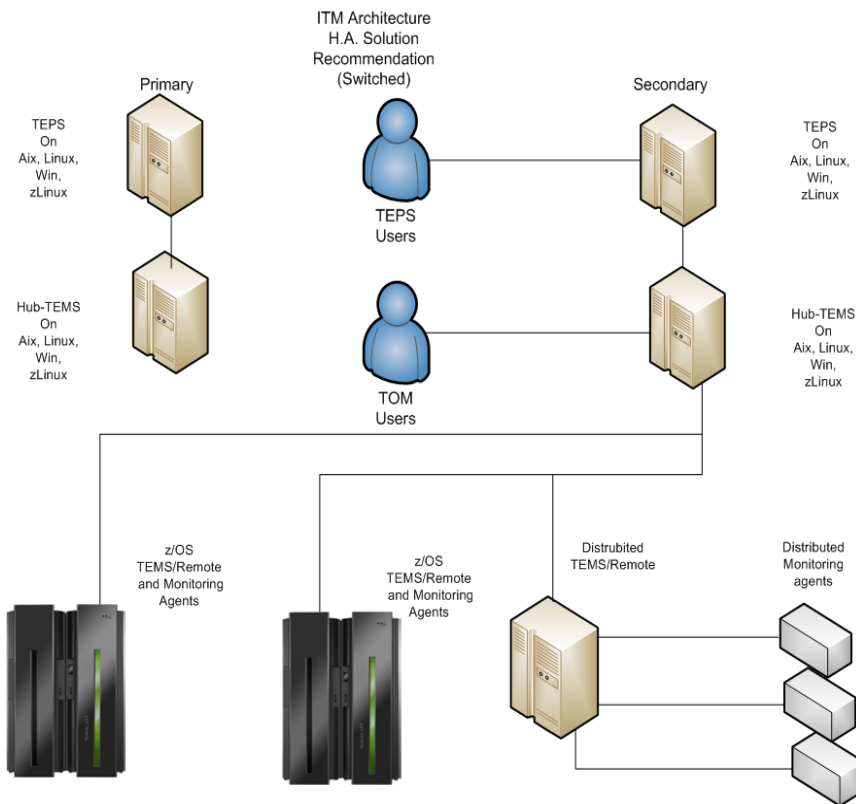
19.11 High Availability / Fail Over

The diagrams below show first the ENTERPRISE DEPLOYMENT followed by the switched fail over ENTERPRISE View.



H.A. Recommended Architecture switched after fail over

IBM z Systems **IBM Monitoring – OMEGAMON Best Practices & Performance Guidelines V2**



20. Product Codes

The list in Table 16-1 might not be all-inclusive of the internal component codes for former Candle® products. The code is prefixed with the letter K.

Table 16-1 Candle product codes

Product name	CODE
OMEGAMON/DE for OS/400®	A4
AF/OPERATOR®	AO
AF/R EMOTE® on Windows NT®	AT
Integrated Resource Manager	AU

IBM z Systems **IBM Monitoring – OMEGAMON Best Practices & Performance Guidelines V2**

OMEGAMON 3270 CUA CICS	C2
OMEGAMON/DE for CICS on z/OS	C5
Configuration tool (z/OS Installer)	CI
CandleNet Portal Server	CJ
OMEGAMON DE	CJ
OMEGAMON/DE for CICS on z/OS	CP
OMEGAMON 3270 CUA DB2	D2
OMEGAMON/DE for DB2 on z/OS	D5
OMEGAMON 3270 CUA Storage	DF
OMEGAMON/DE for DB2plex	DP
Candle Management Server	DS
Candle Dump Analysis Tool on z/OS	DZ
SNMP Gateway on Windows NT	EN
OMEGAMON 3270 CUA IMS	I2
OMEGAMON 3270 CUA 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/DE for Linux	LZ
OMEGAMON 3270 CUA Z/OS	M2
OMEGAMON/DE on z/OS	M5
CCC for MQSeries Configuration	MC
PQEDIT/Z/OS	MD
OMEGAMON/DE for WebSphere MQ	MQ
OMEGAVIEW	MV



Candle Management Workstation	MW
OMEGAMON/DE for Mainframe Networks	N3
OMEGAMON/DE for Windows NT/2000	NT
Alert Adapter for TME® 1 0TM NetView	NV
OMEGAMON/DE for Netware	NW
OMEGAMON 3270 CUA DB2 Classic	O2
OMEGAMON Base (Component of OMEGAMON Classics)	OB
OMEGAMON 3270 CUA CICS Classic	OC
OMEGAMON/DE for UNIX System Services (USS)	OE
OMEGACENTER GATEWAY for Z/OS	OG
OMEGAMON 3270 CUA IMS Classic	OI
OMEGAMON 3270 CUA Z/OS Classic	OM
OMEGAMON 3270 CUA Mainframe Networks	ON
OMEGAMON/DE for MS SQL Server	OQ
OMEGAMON/DE for Oracle	OR
OMEGAMON/DE for Sysplex	OS
OMEGAMON for VM	OV
OMEGAMON/DE for Sybase	OY
OMEGAMON/DE for WebSphere MQ Integrator	QI

Product name	CODE
AF/REMOTE on Windows NT	RP
OMEGAMON/DE for Storage	S3
OMEGAMON/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 IBMi Enterprise Console	TV



DB/WORKBENCH (OK to ship standalone)	TW
OMEGAMON/DE for Tuxedo	TX
OMEGAMON/DE for DB2 Universal DatabaseTM	UD
Universal Agent	UM
OMEGAMON/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



21. Notices

IBM can 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.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

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 might not apply to you.

This information could include technical inaccuracies or typographical errors.

Changes are periodically made to the information herein; these changes are incorporated in new editions of the publication. IBM can make improvements and/or changes in the products and/or the programs 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 can use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
1 New Orchard Road
Armonk, New York 10504-1722
United States

Such information can be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM International Program License Agreement or any equivalent agreement between us.

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

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating systems. You can 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 system 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 can 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.

If you are viewing this information in softcopy form, the photographs and color illustrations might not Web.

21.1 Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com) are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at www.ibm.com/legal/copytrade.shtml



IBM z Systems **IBM Monitoring – OMEGAMON Best Practices & Performance Guidelines V2**

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

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

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

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

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