User's Guide

OMEGAMON® XE for WebSphere Application Server for OS/390

Version 110

Document Number WW54-6554-3

September 2002

Candle Corporation
201 North Douglas Street
El Segundo, California 90245-9796

Candle®

Threaded Environment for AS/400, Patent No. 5,504,898; Data Server with Data Probes Employing Predicate Tests in Rule Statements (Event Driven Sampling), Patent No. 5,615,359; MVS/ESA Message Transport System Using the XCF Coupling Facility, Patent No. 5,754,856; Intelligent Remote Agent for Computer Performance Monitoring, Patent No. 5,781,703; Data Server with Event Driven Sampling, Patent No. 5,809,238; Threaded Environment for Computer Systems Without Native Threading Support, Patent No. 5,835,763; Object Procedure Messaging Facility, Patent No. 5,848,234; End-to-End Response Time Measurement for Computer Programs, Patent No. 5,991,705; Communications on a Network, Patent Pending; Improved Message Queuing Based Network Computing Architecture, Patent Pending; User Interface for System Management Applications, Patent Pending.

NOTICE: This documentation is provided with RESTRICTED RIGHTS. Use, duplication, or disclosure by the Government is subject to restrictions set forth in the applicable license agreement and/or the applicable government rights clause.
This documentation contains confidential, proprietary information of Candle Corporation that is licensed for your internal use only. Any unauthorized use, duplication, or disclosure is unlawful.

Contents

List of Tables

List of Figures

Preface

This guide provides a fundamental understanding of how to use Version 110 OMEGAMON® XE for WebSphere Application Server for OS/390 to monitor availability and performance of the IBM WebSphere Application Server 4.0.*x* on the OS/390 and z/OS platforms.

This guide is designed to help you

- understand how OMEGAMON XE for WebSphere Application Server for OS/390 works with the front end that is provided by CandleNet Portal® (CNP)
- access specific information about events in your environment
- use the real-time data, alerts, historical data, and automation that OMEGAMON XE for WebSphere Application Server for OS/390 provides as a management tool
- use OMEGAMON XE for WebSphere Application Server for OS/390 to define and track instrumented workloads for performance analysis

Note: *This product includes software developed by the Apache Software Foundation (http://www.apache.org/). The Apache Software License, Version 1.1, copyright © 2001, the Apache Software Foundation. All rights reserved.*

About This Book

Who should read this book

This guide is intended to be used by administrators and application programmers for IBM WebSphere Application Server for z/OS and OS/390. It is also intended for system experts who perform a variety of tasks such as performance tuning and troubleshooting.

This manual assumes that you

- have installed CandleNet Portal
- have installed OMEGAMON XE for WebSphere Application Server for OS/390
- are familiar with basic CandleNet Portal concepts, tasks, and features

If you are unfamiliar with these concepts and tasks, please review the *Administering OMEGAMON Products: CandleNet Portal* manual. You should also review the online CandleNet Portal Tour to become familiar with the product's features and capabilities.

This manual also assumes that the IBM WebSphere Application Server for z/OS and OS/390 is installed and that you are familiar with its basic concepts. Refer to the IBM WebSphere Application Server for z/OS and OS/390 documentation for details.

Documentation set information

Candle® provides this single User's Guide for specific OMEGAMON XE for WebSphere Application Server for OS/390 documentation. It introduces you to the features, reports, attributes, and product-provided solutions for the *OMEGAMON XE for WebSphere Application Server for OS/390*.

The documentation for the CandleNet Portal and CandleNet Command Center® products also provides supplementary documentation.

Where to look for more information

For more information related to this product and other related products, please see the

- technical documentation CD-ROM that came with your product
- technical documentation information available on the Candle web site at www.candle.com
- online Help provided with this and the other related products.

Ordering additional documentation

To order additional product manuals, contact your Candle Support Services representative.

We would like to hear from you

Candle welcomes your comments and suggestions for changes or additions to the documentation set. A user comment form, located at the back of each manual, provides simple instructions for communicating with the Candle Information Development department. You can also send email to UserDoc@candle.com. Please include "OMEGAMON XE for WAS for OS/390 User’s Guide, V110" in the subject line.

Adobe Portable Document Format

Printing this book

Candle supplies documentation in the Adobe Portable Document Format (PDF). The Adobe Acrobat Reader will print PDF documents with the fonts, formatting, and graphics in the original document. To print a Candle document, do the following:

1. Specify the print options for your system. From the Acrobat Reader Menu bar, select **File > Page Setup…** and make your selections. A setting of 300 dpi is highly recommended as is duplex printing if your printer supports this option.
2. To start printing, select **File > Print…** on the Acrobat Reader Menu bar.
3. On the Print pop-up, select one of the **Print Range** options for
   - All
   - Current page
   - Pages from: [ ] to: [ ]
4. (Optional). Select the Shrink to Fit option if you need to fit oversize pages to the paper size currently loaded on your printer.

Printing problems?

The print quality of your output is ultimately determined by your printer. Sometimes printing problems can occur. If you experience printing problems, potential areas to check are:

- settings for your printer and printer driver. (The dpi settings for both your driver and printer should be the same. A setting of 300 dpi is recommended.)
- the printer driver you are using. (You may need a different printer driver or the Universal Printer driver from Adobe. This free printer driver is available at www.adobe.com.)
- the halftone/graphics color adjustment for printing color on black and white printers (check the printer properties under **Start > Settings > Printer**). For more information, see the online help for the Acrobat Reader.
- the amount of available memory in your printer. (Insufficient memory can cause a document or graphics to fail to print.)

For additional information on printing problems, refer to the documentation for your printer or contact your printer manufacturer.

Contacting Adobe

If additional information is needed about Adobe Acrobat Reader or printing problems, see the Readme.pdf file that ships with Adobe Acrobat Reader or contact Adobe at www.adobe.com.

Documentation Conventions

Introduction

Candle documentation adheres to accepted typographical conventions for command syntax. Conventions specific to Candle documentation are discussed in the following sections.

Panels and figures

The panels and figures in this document are representations. Actual product panels may differ.

Required blanks

The slashed-b (b̸) character in examples represents a required blank. The following example illustrates the location of two required blanks.

```
b̸eBA*ServiceMonitorb̸0990221161551000
```

Revision bars

Revision bars (|) may appear in the left margin to identify new or updated material.

Variables and literals

In examples of command syntax, uppercase letters are actual values (literals) that the user should type; lowercase letters are used for variables that represent data supplied by the user. Default values are underscored.

```
LOGON APPLID (cccccccc)
```

In the above example, you type LOGON APPLID followed by an application identifier (represented by *cccccccc*) within parentheses.

Note: *In ordinary text, variable names appear in italics.*

Symbols

The following symbols may appear in command syntax:

Table 1. Symbols in Command Syntax

<table>
<tr><th>Symbol</th><th>Usage</th></tr>
<tr><td>|</td><td>The “or” symbol is used to denote a choice. Either the argument on the left or the argument on the right may be used. Example:
YES | NO
In this example, YES or NO may be specified.</td></tr>
<tr><td>[ ]</td><td>Denotes optional arguments. Those arguments not enclosed in square brackets are required. Example:
APPLDEST DEST [ALTDEST]
In this example, DEST is a required argument and ALTDEST is optional.</td></tr>
<tr><td>{ }</td><td>Some documents use braces to denote required arguments, or to group arguments for clarity. Example:
COMPARE {workload} -
REPORT={SUMMARY | HISTOGRAM}
The *workload* variable is required. The REPORT keyword must be specified with a value of SUMMARY or HISTOGRAM.</td></tr>
<tr><td>_</td><td>Default values are underscored. Example:
COPY infile outfile - [COMPRESS={YES | NO}]
In this example, the COMPRESS keyword is optional. If specified, the only valid values are YES or NO. If omitted, the default is YES.</td></tr>
</table>

Overview

Introduction

This chapter introduces the OMEGAMON XE for WebSphere Application Server for OS/390 agent and explains how it can help you monitor and administrate your systems that deploy IBM WebSphere Application Server for OS/390. This chapter explains how this OMEGAMON XE agent works and how to use the CandleNet Portal interface.

Chapter contents

Operating Environment

Overview

OMEGAMON XE for WebSphere Application Server for OS/390 can help you address the increasing challenges presented by today's complex e-business. In this unit, you learn about the operating environment in which the OMEGAMON XE for WebSphere Application Server for OS/390 functions.

What is OMEGAMON XE?

OMEGAMON XE is a suite of products that assists you in monitoring your mainframe and distributed systems on a variety of platforms using a variety of workstations.

It provides a way to monitor the availability and performance of all the systems in your enterprise from one or several designated workstations. It provides many useful reports which you can use to track trends and understand and troubleshoot system problems. You can use OMEGAMON XE to

- establish your own performance thresholds
- create situations, which are conditions to monitor
- create and send commands to systems in your managed enterprise by means of the Take Action feature. Take Action enables you to, for instance, initiate the collection of Java Virtual Machine (JVM) statistics or define thresholds of reporting transaction exceptions.
- create comprehensive reports about system conditions
- monitor for alerts on the systems and platforms you are managing
- trace the causes leading up to an alert
- define your own queries, using the attributes provided by OMEGAMON XE for WebSphere Application Server for OS/390, to monitor conditions of particular interest to you

What is OMEGAMON DE?

The OMEGAMON DE feature package for CandleNet Portal offers a dashboard view of your enterprise. This feature provides a single point of control from which you can manage the resources your business-critical applications rely on, including a range of operating systems, servers, databases, mainframes, and Web components.

It enables you to pull together the data and functionality built into all Candle monitoring solutions, such as OMEGAMON XE for WebSphere Application Server for OS/390. OMEGAMON DE provides all the capability of OMEGAMON XE, plus these additional capabilities.

- Multiple applications can be displayed in one workspaces.

  In a single workspace, you can build a table or chart with data from one type of monitoring agent, and another table or chart with data from a different agent. Within that workspace, you can show views from as many different agent types as are included in that branch of the Navigator.

- You can link application workspaces.

  You can define a link from a workspace associated with one type of monitoring agent to a workspace associated with another type of agent.

- You can define enterprise-specific Navigator views.

  The Navigator's physical view shows the hierarchy of your managed enterprise by operating platform and type of Candle agent. The Navigator's business view shows the hierarchy of any managed objects defined through the Candle Management Workstation® (CMW™). You can also define Navigator views for any logical grouping, such as a departmental or site hierarchy.

- You can define a graphic view.

  The graphic view enables you to retrieve and display real-time monitoring data from Candle agents. Using the graphic view, you can create a background image or import one, then place objects (Navigator items) on the image or have it done automatically using geographical coordinates. You can zoom in and add such graphics as floor plans and organization charts.

- You can review information provided by Candle's Universal Integrator.

  Candle's Universal Integrator is an agent you can configure to monitor any data you collect. It lets you integrate data from virtually any platform and any source, such as, custom applications, databases, systems, and subsystems. Your detailed data providers are listed in the Navigator, and default workspaces are automatically created.

If the OMEGAMON DE package is installed, a new tab, called the Business view, appears at the bottom of your Navigator pane when you are using OMEGAMON XE for WebSphere Application Server for OS/390. Detailed information about OMEGAMON DE is provided in the CandleNet Portal online Help.

Components of the environment

The client-server-agent implementation includes

- both a workstation and a browser version of a CandleNet Portal client
- a server, known as the Candle Management Server® (CMS™) and its Candle Management Workstation (CMW) interface
- a CandleNet Portal server that performs common CandleNet Portal functions and serves to lighten the CandleNet Portal client
- monitoring agents that collect and distribute data to a CMS. The OMEGAMON XE for WebSphere Application Server for OS/390 is such an agent.

What is CandleNet Portal?

CandleNet Portal provides a view of your enterprise from which you can drill down to examine more closely components of your systems environment. Its application window consists of a Navigator that shows all the systems in your enterprise where Candle agents are installed, and a workspace that includes table and chart views of system and application conditions.

CandleNet Portal runs situations at regular intervals to check that your applications and system resources are running, and running well. A failed test causes event indicators to appear in the Navigator.

CandleNet Portal offers two modes of operation: desktop and browser. In desktop mode, the application software is installed on your system. In browser mode, the system administrator installs the application on the web server and you start CandleNet Portal from your browser. In browser mode, the software is downloaded to your system the first time you log on to CandleNet Portal, and thereafter only when there are software updates.

What is the Candle Management Workstation?

When using OMEGAMON XE for WebSphere Application Server for OS/390, you use the CMW interface to perform user administrative functions.

Further Information about CMW and CandleNet Portal

For further information about CandleNet Portal, refer to the product's Administrator's Guide and its online Help.

Further information about using the CMW for user administration can be found in the CMW Administrator's Guide as well as in the CMW online Help.

If you are an experienced CMW user, you should review the CandleNet Portal Help topic called *"CMW Users."* This topic addresses the similarities and differences of these user interfaces, provides guidance about each interface's functionality, and a cross-reference listing that clarifies the products' terminology.

Understanding Attributes

Using attributes

OMEGAMON XE gathers data from remote agents residing on the managed systems of your network and represents this data in system elements called attributes.

You can use these attributes to build situations to monitor the performance of the underlying WebSphere Application Server for z/OS and OS/390.

To use these attributes effectively, you must first understand the structure of an attribute.

- An attribute is a discrete characteristic or piece of information about a managed system
- An attribute belongs to an attribute group which includes attributes that are related.
- An attribute group associates attributes that refer to some unique characteristic of the data that is being stored.
- An attribute item stores data for a particular property of an attribute group.

For example, the OMEGAMON XE for WebSphere Application Server for OS/390 attribute

<Application_Server_Instance>.<Garbage_Collection_Interval>

<Garbage_Collection_Interval> is an attribute item that reports the amount of time spent on garbage collection. This attribute belongs to the <Application_Server_Instance> attribute group.

Available attribute groups

Version 110 of OMEGAMON XE for WebSphere Application Server for OS/390, provides 26 attribute groups to provide configuration, status, and performance information for your WebSphere environment.

For more information on attributes

For complete descriptions of the attributes for OMEGAMON XE for WebSphere Application Server for OS/390, refer to the online Help for OMEGAMON XE for WebSphere Application Server for OS/390.

Introduction to Situations

What is a situation?

A situation is a logical expression involving one or more system conditions. Situations are used to monitor the condition of systems in your network. They can be used to monitor important aspects of the managed systems in your enterprise. You can manage situations from CandleNet Portal by using the Situation editor.

What is a predefined situation?

The agents that monitor your system environment are shipped with a set of predefined situations that you can use as-is or modify to meet your requirements.

Predefined situations give you out-of-the-box experience with Candle products. They contain attributes and expressions that check for system conditions common to many enterprises. These situations can issue alerts that notify you about conditions or events that you may want to track.

Using predefined situations can improve the speed with which you can begin using the OMEGAMON XE for WebSphere Application Server for OS/390. You can examine and, if necessary, change the conditions or values being monitored by a predefined situation to those best suited to your enterprise.

Note: *Candle suggests that, if you choose to modify a predefined situation, you first make a copy to ensure fallback if necessary.*

Using situations

You manage situations from the CandleNet Portal GUI by using the Situation editor. Using the Situation editor you can

- create a situation
- save a situation
- display a situation
- edit a situation
- start, stop, or delete a situation

- investigate the event workspace for a situation

When you open the Situation editor, its left frame initially lists the situations associated with the Navigator item you selected. When you click a name of a situation or create a new situation, the right frame of the Situation editor opens to provide you with the following information about the situation or to let you further define that situation:

Condition Check, add to, and edit the condition being tested.

Distribution Check the systems to which the situation is assigned and assign the situation to systems.

Expert Advice Write comments or instructions to be read in the event workspace.

Action Specify a command to be sent to the system.

You can also enter take action commands by adding a take action view to a workspace, selecting Take Action from the pop-up menu for an item in the Navigator's physical view, or creating take action commands and saving them for later use.

Until Reset a true situation when another situation becomes true or a specified time interval elapses.

Alerts for OMEGAMON XE for WebSphere Application Server for OS/390

The following situations issue alerts based on the conditions defined in the situations.

- The WAS390_Application_Server_Error situation issues a critical alert if the Product Events for WAS OS/390.Severity attribute reports an error.
- The WAS390_AppSrvr_Instance_Not_Up situation issues a critical alert if the Application Server Instance.Status attribute does not equal one of the following values: Running, Initializing, or n/a.

  The alert means that the application server is not running. If you want to start the server, use the Take Action view and enter the necessary MVS console commands.
- The WAS390_SrvrInstance_Not_Running situation issues a critical alert if the Server Instance Status.Status attribute does not equal one of the following values: Running, Initializing, or n/a.

This alert indicates the application server is not running. Start the server if needed. To start the server, use the Take Action view and enter the necessary MVS console commands.

Further information

For the most current information about situations, refer to the Administrator's Guide or the online Help provided with CandleNet Portal.

Take Action Commands

Introduction

OMEGAMON XE for WebSphere Application Server for OS/390 provides several predefined take action commands, which can be initialed from CandleNet Portal. The take action commands for this product (commands sent to this product's agent) have names that begin with WW.

You can invoke the commands presented here from the Take Action view, from a situation (when it becomes true), from the Navigator, or from a row in a table view. For more information on how to send a commands, refer to the Help for CandleNet Portal.

Prerequisite

You must include the TakeActionAuthUsers parameter in the KWWXML configuration file in order to enable take action commands. No one can issue take action commands unless this parameter is specified.

Who can issue take action commands

You can configure the OMEGAMON XE for WebSphere Application Server for OS/390 so that only a designated set of users can successfully issue its take action commands. Authorized take action users can be specified

- at the agent level, using the TakeActionAuthUsers parameter
- at the monitored application server level, using the TakeActionAuthUsers parameter

See "TakeActionAuthUsers" on page 56 for details.

Command restrictions on OS/390 and z/OS

If you find it necessary to issue a command that is not supported by the agent, you can issue it as a system command (OS/390 console command) instead. Use the exact syntax of the command as documented by IBM. The OS/390 console command does not require the WW: prefix.

Take action commands provided for OMEGAMON XE agent

Each of the take action commands provided with the OMEGAMON XE for WebSphere Application Server for OS/390 XE agent is described below. The Help for OMEGAMON XE for WebSphere Application Server for OS/390 explains how to complete the Edit Argument Values dialog box with the parameter values that are needed to control the operation of the commands. This User's Guide also contains additional information for using the take action commands as indicated by the cross-references for the commands.

WAS390 JVMSTAT - J2EE

Use the WAS390 JVMSTAT command to start or stop JVM profiling for the server and also control the level of profiling that is to be performed. When this command is received, JVM profiling for the specified application is started. If the JVMProfilerCollect parameter is empty, then the command stops JVM profiling upon its receipt. This command can be targeted at a particular application server.

The syntax of the command is

WW:JVMSTAT JVMProfilerCollectStopAfter=<*profiling should stop after this amount of time in seconds*>

JVMProfilerCollect=<*type of data to collect*>

JVMProfilerCPUSampleInterval=<*CPU sampling interval*>

Frequency=<*profiler sampling interval*>

Classes=(<*list of class masks*>)

After selecting the command, click the Arguments button to display the Edit Argument Values pop-up. See the Help for the agent's take action commands and "Activating the JVM Profiler" on page 66 for details about issuing this command and supplying the required parameter values.

WAS390 Dynamic Workload Analysis

Use the WAS 390 Dynamic Workload Analysis command to enable or disable the collection of workload analysis data after a class file has been loaded. You can issue multiple CollectWorkload commands. The effect of multiple commands is cumulative with the operands on one command being applied with the operands of all currently active CollectWorkload commands.

The syntax of the command is

> **WW:CollectWorkload ClassType=<*list of class types*>**
> **ClassName=<*list of class name masks*>**

After selecting the command, click the Arguments button to display the Edit Argument Values pop-up. See the Help for the agent's take action commands and "Setting Up the Workload Analysis Control File" on page 72 and "Workload Information for Performance Tuning" on page 157 for details about setting up your system and using workload analysis data collection.

Note: *To disable the collection of all workload analysis data, issue a CollectWorkload command with no ClassType or ClassName parameters.*

WAS390 Set Workload Exception Thresholds

This command defines the thresholds for the collection and reporting of transaction exception data.

The syntax of the command is

> **WW:SetExceptionWorkload Name=<*workload-name*>**
> **Max=<*maximum number to collect*>**
> **MinRespTime=<*minimum response time to report*>**

After selecting the command, click the Arguments button to display the Edit Argument Values pop-up. See the Help for the agent's take action commands and "Setting Up the Workload Analysis Control File" on page 72 and "Workload Information for Performance Tuning" on page 157 for details about setting up your system and using workload exceptions data collection.

Note: *To disable the collection of all transaction exception data, issue a SetExceptionWorkload command with no Name, Max, or MinRespTime parameters*

WAS390 Start Application Tracing

Use this command to start the application trace facility that traces the flow within instrumented methods. This command creates a trace file and begins recording to it. The trace will show each entry and exit for each instrumented Java method. The tracing stops and this file is closed when either the amount of time you set for the trace or the number of trace entries is reached. Once the trace has been turned off, the OMEGAMON XE agent can read the trace file and display the entries in the Application Trace Files and Application Trace workspaces.

The syntax of the command is

WW:StartAppTrace WorkloadType={Servlet | EJB | Both}
ClassNames=<*list of class name masks*>
MaxEntries=<*maximum entries to collect*>
MaxTime=<*maximum amount of time*>

After selecting the command, click the Arguments button to display the Edit Argument Values pop-up. See the Help for the agent's take action commands and "Setting Up the Workload Analysis Control File" on page 72 for details about starting an application trace.

WAS390 Stop Application Tracing

Use this command to stop the current application trace.

The syntax of the command is

WW:StopAppTrace

See the Help for the agent's take action commands and "Setting Up the Workload Analysis Control File" on page 72 for details about stopping an application trace.

WAS390 Delete Application Tracing Files

Use this command to manage (delete) application trace files.

The syntax of the command is

WW:DeleteTrace ID=<*trace ID*>

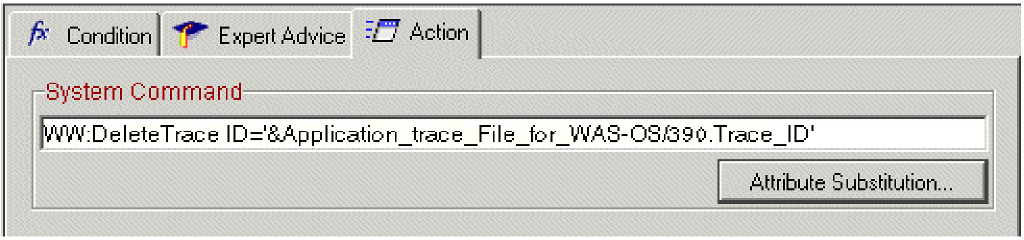
where <*trace ID*> is the numeric identifier that was assigned to the trace file. See the Help for the agent's take action commands for details about using this command.

Using take action commands in situations

You can use take action commands from within predefined situations and in situations that you write. This feature is built into OMEGAMON XE products and is explained in the Help for CandleNet Portal and in the *Administering OMEGAMON Products: CandleNet Portal* guide. The term, reflex automation, that is used in these sources refers to a situation that issues a command.

To issue a take action command, click the **Action** tab within the situation editor and complete the System Command field. Prefix the command with **WW:** and follow it with the Take Action command you want to use. Then, click the **Attribute Substitution** button to add an attribute to the command. Enclose this attribute in single quotes.

This example shows the syntax for the DeleteTrace take action command.



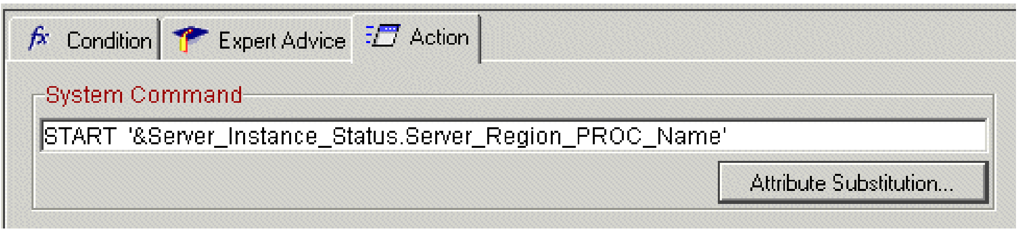
```
WW:DeleteTrace ID='&Application_trace_File_for_WAS-OS/390.Trace_ID'
```

Using system commands in situations

You can use issue system commands from within situations. This feature is explained in greater detail in the Help for CandleNet Portal and in the *Administering OMEGAMON Products: CandleNet Portal* guide.

To issue a system command, click the **Action** tab within the situation editor and complete the System Command field. Click the **Attribute Substitution** button to add an attribute to the command.

The example below issues a START command to start a server instance. It places the Server Region PROC Name attribute obtained from the list of attributes in single quotes.



```
START '&Server_Instance_Status.Server_Region_PROC_Name'
```

The example used the WAS390_AppSrv_Instance_Not_Up situation, which notifies you when the status of the instance is found not to be in a Running, Initializing, or n/a state.

Understanding OMEGAMON XE for WebSphere Application Server for OS/390 Information

Workspaces and table views

When using CandleNet Portal, information is displayed in workspaces. Within a given workspace, information may be displayed in tabular form. CandleNet Portal refers to this tabular format for information as a table view. Information may also be displayed in the workspace as a chart, such as a pie chart or bar graph, or other format you can specify.

What is a workspace?

A workspace is the working area of the CandleNet Portal application window. At the left of the workspace is a Navigator that you can use to select the workspace you want to display. As part of the application window, the right side of the top status bar shows the CandleNet Portal server name and port number to which the displayed information applies, as well as the ID of the current user.

As you select items in the Navigator, the workspace presents views pertinent to your selection. Each workspace has at least one view. You can customize a workspace is to change the type of view or to add views to the workspace.

Defining workspace properties

Every workspace has a set of properties associated with it. You can customize the workspace by working in the Properties editor to change the style and content of each view.

The properties of a workspace may be some or all of the following

Query Specify what data should go in the chart or table.

Filters Refine the view by filtering out unwanted data from the chart or table.

Thresholds Establish threshold values and color indicators for a table view.

Configuration Specify the script to run or the connection to make whenever you open the terminal view.

Style Change the behavior and appearance of the view.

Changes you make to a workspace, such as adding or editing a view are only temporary. They will be lost when you exit CandleNet Portal unless you save the workspace.

Viewing information in CandleNet Portal

The CandleNet Portal GUI can present information in any of the views below:

- Table view
- Pie chart view
- Bar chart view
- Plot chart view
- Needle gauge view
- Thermometer gauge view
- Notepad view
- Enterprise Event Console view, that shows the status of the situations associated with the system
- Take Action view, that is used to send a command to the system
- Terminal view, that enables you to start a 3270 or 5250 work session
- Browser view, that permits you to open a browser to see HTML pages and Web sites

Associating workspaces with attributes

There is a direct relationship between attributes and views, in that information presented in a view, such as a table view or chart, is always from a single attribute group. Column names in a table view typically correspond to attribute names.

For more information

Continue reading to learn how to manipulate information presented to you by CandleNet Portal. For descriptions of the individual workspaces, see "Introducing OMEGAMON XE for WebSphere Application Server for OS/390 Information" on page 83.

Using OMEGAMON XE for WebSphere Application Server for OS/390 information

You can view information about each managed system that you are monitoring. Use this information to

- monitor the performance of each managed system, helping you to identify system bottlenecks and evaluate tuning decisions
- check the status of server instances
- gather performance data reported by the Systems Management Facility (SMF)
- monitor response time for workloads you have defined to analyze performance
- set up traces to collect key diagnostic information
- select the most effective threshold values for situations you create
- review status information when a change in the state of a given resource occurs; such as from OK to Warning or Warning to Critical

Investigating an event

When the conditions of a situation have been met, the situation evaluates True, causing an event indicator to appear in the Navigator. You can investigate the event by opening its workspace.

The event workspace shows two table views, one with the attributes' values when the situation evaluated True, and the other with the attributes' current values.

The event workspace can also display a text view with any expert advice written by the situation's author, and the Take Action view so you can send a command to a managed system.

Adding a workspace to your favorites

When using CandleNet Portal in browser mode, you can start it from any workstation by entering the URL for the web server where the browser mode client is installed. Each CandleNet Portal workspace also has a URL so that you can save the workspace to your Favorites list or specify it as your home page.

Filtering information

To manually set up filtering for a given table view, place the cursor on the table view, press the right mouse button, and select **Properties.** From the displayed dialog, select the Filters tab. Here you can select the columns to display as well as set up the criteria to filter rows to display. To save your filtering specifications, you must save the workspace before exiting.

Sorting Information

Sorting is handled by simply clicking on a column heading. Click once and the report will be sorted in ascending order. Click a second time to resort the report into descending order. A third click restores the report’s default sort.

2

Configuring OMEGAMON XE for WebSphere Application Server for OS/390

Introduction

This chapter discusses environment variables and configuration parameters that are used for OMEGAMON XE for WebSphere Application Server for OS/390. It also explains the settings and procedures you need to use to collect Java Virtual Machine (JVM) statistics and set up instrumentation to collect workload analysis data.

Chapter contents

Background Information

Chapter goals

This chapter provides configuration information that enables your to modify the operation of the OMEGAMON XE for WebSphere Application Server for OS/390 agent. It contains

- basic configuration information about the OMEGAMON XE agent for WebSphere Application Server
- information about environment variables that affect the operation of Candle Technologies (CT) environment that supports the product
- information about using the JVM profiler to collect JVM statistics for application servers
- information about configuring your agent and IBM's WebSphere Application Server to collect workload analysis and application trace data

Prerequisites

This chapter assumes that you have performed the following operations.

- unloaded the product tape and completed the CICAT installation of the OS/390 and z/OS product components and the OMEGAMON XE agent.

  Refer to the installation documentation that accompanied this product for complete details.
- used the WebSphere Application Server for OS/390 Administrative Console to establish that user ID for starting the agent PROC for OMEGAMON XE for WebSphere Application Server for OS/390 has WebSphere Application Server administrator authority and also DB2 permissions. (The agent PROC name has been added as a user ID to UNIX System Services (USS).)
- established that the agent PROC owner has read and write authority to the Candle *rhilev.**.** datasets.
- used the product CD-ROM to install CandleNet Portal and the OMEGAMON XE client for WebSphere Application Server for OS/390

Setup information for the WebSphere Application Server for OS/390

Complete the following setup procedures in WebSphere Application Server for OS/390 to support the Candle product.

1. Create a WebSphere administrator ID for the agent STC. Refer to “Define administrator identities” in the WebSphere Application Server V4.0.1 for z/OS and OS/390 System Management User Interface.
2. Give the new administrator the same RACF authorities as the CBADMIN ID.
3. Add the new administrator to the access control list in LDAP. Refer to “Steps for updating the access control list for LDAP” in the *WebSphere Application Server V4.0.1 for z/OS and OS/390 Installation and Customization Guide*.
4. Grant the new administrator authority to deploy a J2EE application in the system management database. Refer to “Steps for granting the administrator database authorities” in the *WebSphere Application Server V4.0.1 for z/OS and OS/390 Installation and Customization Guide*.

Chapter navigator

The chart below shows where you can locate information about specific tasks you want to perform.

If you want to...	**See...**
View the default configuration file,	page 44
Review the basic configuration parameters in the KWWXML member that are set during installation,	page 46
View the configuration file that contains optional parameters,	page 48
Review optional configuration parameters in the KWWSAXML member,	page 49
Obtain details about each KWWAGENT configuration parameter,	page 44
Obtain details about each MonitorAppServer configuration parameter,	page 56
Obtain details about each environment variable,	page 64
Learn how to activate the JVM profiler to collect JVM statistics,	page 66
View the workload analysis control file in which you define your workloads,	page 73
Configure WebSphere Application Server for OS/390 for workload analysis and application trace data	page 78

Setting up RACF Authorizations

Background

If you use RACF to protect your system resources, then several authorizations must be established to enable the OMEGAMON XE for WebSphere Application Server for OS/390 to function normally. Complete the authorizations described below.

Authorization for the MVSADMIN.WLM.POLICY resource

The userid that is associated with the KWWAGENT address space must have READ authority to the MVSADMIN.WLM.POLICY resource if the resource is defined to RACF. Use the following RACF command to grant this authority:

```
PERMIT MVSADMIN.WLM.POLICY CLASS(FACILITY)
                           ID(agent-userid)
                           ACCESS(READ)
```

Authorization for the logstream resource

The userid that is associated with the KWWAGENT address space must have READ authority to the logstream resource, if the resource is defined to RACF. Use the following RACF command to grant this authority:

```
PERMIT logstream-name CLASS(LOGSTRM)
                      ID(agent-userid)
                      ACCESS(READ)
```

where logstream-name is obtained from the LOGSTREAM name parameter in the application server's environmental variables.

Authorization for the virtual console resource

The user ID that is associated with the KWWAGENT address space must have READ authority to the virtual console RACF resource, which may need to be defined to RACF. Use the following RACF command to grant this authority:

```
RDEFINE OPERCMDS MVS.MCSOPER.KWW* UACC(NONE)
PERMIT MVS.MCSOPER.KWW* CLASS(OPERCMDS)
                        ID(agent-userid)
                        ACCESS(READ)
```

User ID authorizations

The user ID that is associated with the WebSphere Application Server Server Instance server region must also have certain authorizations. This user must have the ability to read and write into the Application Trace directory, as specified by the AppTraceDirectory keyword in the KWWXML member. The user ID must also have read authority to the files contained in the /CANDLEHOME/kww directory.

Configuration Basics about the OMEGAMON XE Agent

Overview

The intelligent remote agent (IRA) for OMEGAMON XE for WebSphere Application Server for OS/390 is called KWWAGENT. This agent uses configuration information and environment variables to direct its operation and monitoring characteristics. Their default settings are established when you install OMEGAMON XE for WebSphere Application Server for OS/390. This chapter discusses these configuration parameters and environment variables, their uses, and the changes you can make to the default installation settings.

Default configuration for the agent

The default configuration values for the KWWAGENT should suit most installations initially. However, you are likely to modify these parameters when you want to enable different types of data collection that are initially turned off to minimize overhead. For example, you will need to change the parameter settings to collect and view information about application trace files and view the data within an application trace or to report Java Virtual Machine (JVM) data from the Candle profiler.

Configuration parameters for the agent

The initial parameters and their settings are defined in the dataset member, KWWXML, which is created on the mainframe when you install OMEGAMON XE for WebSphere Application Server for OS/390. These default parameters are located in the *rhilev.midlev.*RKANDATV(KWWXML). The name of this member is specified in *rhilev.midlev.*RKANCMD(KWWAGST).

Figure 1 on page 47 contains an example of the KWWXML dataset member and its default settings. This chapter also contains a sample of additional optional configuration parameters that are defined in the dataset member, *rhilev.*TKANSAM(KWWSAXML). This member is created on the mainframe when you install OMEGAMON XE for WebSphere Application Server for OS/390. See Figure 2 on page 49 for examples of its contents. For descriptions about all possible configuration variables, their use, values, and default settings, see "Your Agent's Configuration Parameters and XML Tags" on page 50.

Environment variables for the agent

OMEGAMON XE for WebSphere Application Server for OS/390 uses IRA environment variables that affect the operation of its KWWAGENT agent. These variables are used by the CT environment that supports the Candle agent. For a detailed discussion of the variables used by the KWWAGENT, see "Your Agent's Environment Variables" on page 64.

A Look Inside Your Agent’s Configuration Files

Introduction

This topic presents two examples found in the *rhilev.midlev.*RKANDATV dataset‘. One is of the dataset KWWXML, which contains the default parameters for the OMEGAMON XE for WebSphere Application Server for OS/390 agent. The other example is the member, KWWSAXML. This member includes sample settings for all the optional configuration parameters that you might want to use after your initial installation.

What are the KWWXML default parameters?

The default parameters that are installed for OMEGAMON XE for WebSphere Application Server for OS/390 contains a base set of values that automatically apply the product defaults when you install the product. The defaults in the KWWXML member should suit most sites initially. From time to time, you are likely to change these defaults to meet your specific needs and tailor the type and amount of data that the agent monitors and reports.

Your initial default configuration dataset member, KWWXML will look like the sample in Figure 1 on page 47, which shows sample values. If you used the installation defaults, you will find the member in *rhilev.midlev.*RKANDATV(KWWXML). For a detailed description of these parameters and other optional parameters, see “Your Agent’s Configuration Parameters and XML Tags” on page 50.

FIGURE 1. Sample Default Configuration Parameters in KWWXML

```
<KWWAGENT
   Version="110"
   JVMProfilingPort="41283"
   WASAppServerRoot="/WAS401/usr/lpp/WebSphere"
   JavaRoot="/WAS401/usr/lpp/java/IBM/J1.3"
   AppTraceDirectory="/u/flivs"
   WWHFSRoot="/u/vheat/ft/kww">
   DefaultJVMProfilerSampleInterval="60"
   DefaultCollectWorkloadClassType="A"
   DefaultCollectWorkloadClassName="*"
</KWWAGENT >
```

Additional start-up configuration parameters

Some of the settings in above example are not defaults that are true for all users. During installation, you are prompted by CICAT to provide values for the following configuration parameters:

- JVMProfilingPort,
- WASAppServerRoot
- JavaRoot
- AppTraceDirectory
- WWHFSRoot

Whatever values you provide to CICAT are then copied into the default KWWXML file. The example in Figure 1 shows sample settings that CICAT would place in KWWXML.

Additional configuration parameters

The default configuration dataset member, KWWXML, defines basic required values for the configuration parameters. In addition to this default member, the installation of OMEGAMON XE for WebSphere Application Server for OS/390 also creates a sample configuration member, KWWSAXML, that shows the usage of all the configuration parameters that you may want to specify for your installation.

About the sample optional configuration parameters

The configuration parameters in *rhilev.midlev.*TKANSAM(KWWSAXML) are given as XML keywords for two XML tags. The parameters for KWWAGENT XML tag govern the agent as a whole. The parameters for the MonitorAppServer tag govern a specific application server instance.

Maintain the XML syntax as shown in Figure 2, "Expanded Agent Configuration Parameters in KWWSAXML," on page 49. The white space and division into separate lines, however, is not important.

FIGURE 2. Expanded Agent Configuration Parameters in KWWSAXML

```
<!--KWWAGENT Configuration Settings-->
<KWWAGENT
     Version="110"
     AgentId=""
     DiscoverInterval="900"
     JVMProfilingPort="65535"
     NamingServerPort="900"
     RetainProductEvents="5"
     WASAppServerRoot="/usr/lpp/WebSphere"
     JavaRoot="/usr/lpp/java/IBM/J1.3"
     WWHFSRoot="/candle/kww"
     LogScanInterval="60"
     RetainLogFileEvents="5"
     DefaultExclude="No"
     DefaultJVMProfilerCollect=""
     DefaultJVMProfilerCollectStopAfter="300"
     DefaultJVMProfilerCollectClasses="*"
     DefaultJVMProfilerCPUSampleInterval="5"
     DefaultJVMProfilerSampleInterval="60"
     DefaultJVMProfilerIgnoreMethods="java/net/SocketInputStream.socketRead"
     DefaultMinimumSamplingInterval="10"
     DefaultRetainLogFileEvents="5"
     DefaultStatisticInterval="60"
     TakeActionAuthUsers="*"
<!--Server Instance Settings—>
     <MonitorAppServer
          Name="BBOASR1A"
          SubnodeAliasName="ApplicationServer"
          Exclude="No"
          JVMProfilerCollect=""
          JVMProfilerCollectStopAfter="300"
          JVMProfilerCollectClasses="*"
          JVMProfilerCPUSampleInterval="5"
          StatisticInterval="60"
          TakeActionAuthUsers="*"
       />
</KWWAGENT >
```

Your Agent's Configuration Parameters and XML Tags

Introduction

This topic covers each of the parameters that can be used to configure the KWWAGENT agent. It describes their uses and values and provides their default settings. The following is a complete list of the parameters for the KWWAGENT and MonitorAppServer XML tags and their keyword values.

Note that only a few of these parameters are used in the default configuration shown in the KWWXML dataset member shown in Figure 1 on page 47. The others are optional parameters that you can use to customize the operation of the agent and the type of data it collects.

KWWAGENT

The KWWAGENT XML tag encapsulates all parameters for the agent. The attributes that follow this tag apply to the agent as a whole. They specify monitoring and connection defaults that can be overridden, if desired, at the individual application server level. The following keyword attributes apply to this tag.

AgentId

This keyword gives a very short identifier for the agent. Specify this attribute if there are multiple KWWAGENTs reporting data to the same CMS hub. Each agent needs a different identifier so that the nodes reporting to the same CMS hub do not have the same name. The default agent ID is "" (none); the maximum length is four characters.

AppTraceDirectory

This keyword identifies the root directory where the application trace files are stored. The valid format is a directory name, either fully qualified or relative to the current KWWAGENT directory. If you use the default AppTraceDirectory value (/candle/kww/apptrace), you need to grant read/write authorization to the Server region identity.

Note: *The user ID associated with the WebSphere Application Server J2EE server region needs to have read/write authorization for the AppTraceDirectory.*

DefaultCollectWorkloadClassName

This keyword is used in combination with the DefaultCollectWorkloadClassType parameter to identify those classes for which workload analysis data is to be collected. This keyword can specify multiple masks, separated by commas or semi-colons. The masks can contain the asterisk (*) and question mark (?) wildcard characters.

In order to collect workload analysis data for a class, the name must have a type in the DefaultCollectWorkloadClassName list and a class that matches one of the DefaultCollectWorkloadClassType masks.

DefaultCollectWorkloadClassType

This keyword specifies the type of workload analysis data that will be collected by default for all J2EE application servers. The valid format is an alphanumeric string, maximum 10 characters. Each character, shown in the following list, represents a specific type of workload profiling data that should be collected.

Value	Meaning
A	Indicates that all workload data should be collected
B	Indicates that workload data for EJB-related classes should be collected
D	Indicates that workload data for SQL-related classes should be collected
I	Indicates the workload date of JNDI-related classes should be collected
M	Indicates that workload data for user-defined methods should be collected
N	Indicates that workload data for socket-related classes should be collected
Q	Indicates that workload data for JMS-related classes should be collected
S	Indicates that workload data for servlet classes should be collected
T	Indicates that workload data for transaction-related classes should be collected

DefaultExceptionWorkloadMax

This keyword sets a limit on the number of exception transactions for each workload identified by the DefaultExceptionWorkloadName parameter. These transactions are those that are reported for exceeding the limits you set for your site. These exceptions are displayed in the Longest Running Workloads workspace. The valid format contains a positive number that specifies the maximum number of exception transactions to collect for the workload.

DefaultExceptionWorkloadMinRespTime

This keyword specifies a response time threshold for reporting exception transactions. Any transactions having lower response times than the threshold value are not reported. The valid format contains an positive number that specifies the number of milliseconds.

DefaultExceptionWorkloadName

This keyword identifies the class name or names of the workloads for which exception transactions are to be collected. It applies to all application servers. The valid format is one or more masks the may contain the asterisk (*) and question mark (?) wildcard characters to identify the class or classes on the application servers for which exceptions are to be collected. Separate multiple masks by commas or semi-colons. If this keyword is omitted, workload exception data is not collected for application servers unless specified at the application server level or by a SetExceptionWorkload take action command.

DefaultExclude="Yes|No"

This keyword indicates whether or not the application servers are to be monitored by default. If DefaultExclude is set to a value of Yes, then no application servers are monitored, unless there is an overriding Excluding value specified explicitly for an application server.

DefaultJVMProfilerCollect

This keyword specifies the type of JVM statistical data that will be collected. The value format is a character string of up to 5 characters. Each character represents a specific type of JVM profiling data that should be collected.

Value	Meaning
C	Java monitor contention
H	Java heap usage
T	Thread statistics (implied if C or M options are set)
M	Java method invocation
G	Garbage collection

If the DefaultJVMProfilerCollect parameter is set to an empty string, it indicates that JVM statistical information is *not* to be collected. This is the default setting for the DefaultJVMProfilerCollect parameter.

DefaultJVMProfilerCollectClasses

This keyword indicates the Java classes for which the profiler should collect method, heap, and contention data. It applies to all application servers. You can specify this keyword as a list of one or more names, separated by commas. Each entry can be a mask that can include the asterisk (*) and question mark (?) wildcard characters. Only Java class names that match one of the specified masks will have method, heap, and contention data collected by the profiler. The default for this parameter is (*) for all classes.

DefaultJVMProfilerCollectStopAfter

This keyword specifies the time span the OMEGAMON XE agent will use to collect JVM statistics for an application server before it automatically stops this collection. The specified value is in seconds. If this value is 0, the OMEGAMON XE agent will collect JVM data until you explicitly stop the collection. If this value is not zero, use a value greater than or equal to the value set for DefaultStatisticInterval. The default is 300.

DefaultJVMProfilerCPUSampleInterval

This keyword indicates how often, in milliseconds, CPU sampling should be performed within the JVM. The smaller this value is, the higher the overhead of JVM data collection, but the more accurate that thread and method invocation data will be. The default is 5. This value is only meaningful if the DefaultJVMProfilerCollect keyword contains 'T' or 'M'.

DefaultJVMProfilerIgnoreMethods

This keyword indicates the names of the Java methods for which CPU busy samples are to be ignored by the profiler. On certain platforms, such as, AIX, the profiler interface incorrectly reports a "CPU busy" condition for certain Java methods. This can lead to misleading results in the Java Threads and Java Methods table views. Use this keyword to indicate that the CPU sample should ignore "CPU busy" conditions reported by the JVM profiler interface if the active method matches one of the entries for this keyword.

You can specify this keyword as a list of one or more names, separated by commas. Each entry can be a mask that may include the (*) and (?) wildcard characters. CPU busy conditions reported against all Java method names that match one of the specified masks will be ignored. The default for this parameter is empty.

DefaultJVMProfilerSampleInterval

This keyword provides the default sampling interval, in seconds, for profiler data for each application server. The KWWAGENT will collect profiler data from the application servers based on the time specified. The default is 60.

DefaultRetain History

This keyword determines the length of time, in seconds, that historical data not yet requested should be retained by the agent. The default is 3600 seconds (60 minutes). If history is being collected, this value should be greater that the historical collection interval. If history is not being collected, this value should be set to 0.

DefaultSMFCollectBeans

This keyword determines which beans will have detailed bean-level data collected for them. Bean data is contained in the SMF J2EE Container Interval records. This parameter can be specified as a list of one or more names or masks, separated by commas. Each entry may include the asterisk (*) and question mark (?) wildcard characters.

DefaultSMFCollectClasses

This keyword determines which classes will have detailed, class-level data collected for them. Class data is contained in the SMF MOFW Container Interval records. This parameter can be specified as a list of one or more names or masks, separated by commas. Each entry may include the asterisk (*) and question mark (?) wildcard characters.

DefaultSMFCollectMethods

This keyword determines which methods will have detailed method-level data collected for them. Methods data is contained in the SMF MOFW and J2EE Container Interval records. This parameter can be specified as a list of one or more names or masks, separated by commas. Each entry may include the asterisk (*) and question mark (?) wildcard characters.

DefaultStatisticInterval

This keyword is reserved for future use.

DiscardSMFRecords

This keyword determines whether or not the SMF records that the agent processed for WebSphere Application Server will should be discarded before being written to the SMF dataset. The default is No.

DiscoverInterval

This keyword specifies how often, in seconds, KWWAGENT will poll the WebSphere Administrative Server for OS/390 to search for newly-defined application servers. The default is 900 (every 15 minutes).

JVMProfilingPort

This keyword indicates the number of the IP port that the individual monitoring agent libraries will use to make contact with KWWAGENT. The default is 65535.

LogScanInterval

This keyword specifies how often, in seconds, KWWAGENT will browse each error logstream in order to detect error or exception conditions. Any new entries found that are associated with Application Servers that Candle is monitoring will be added to the application server error log table as pure events. The default is 60.

RetainLogFileEvents

This keyword specifies the number of log error events to retain for viewing in reports. These events are always reported as pure events for situations, and, if historical situations are active, they can be viewed historically. However, the most recent *n* number of events are seen in the regular reports where *n* is the count given by this keyword. This keyword must be greater than 0. The default is 50.

RetainProductEvents

This keyword gives a count of how many product events to retain as a total for viewing in reports. These events are always reported as pure events for situations, and if historical situations are active, they can be viewed historically. However, the most recent *x* number of events can be viewed in regular reports where *x* is the count given by this keyword. This keyword must be greater than 0. The default is 5. The maximum value is environment dependent.

TakeActionAuthUsers

This keyword indicates which CandleNet Portal (CNP) users are authorized to issue take action commands. Note that there can be multiple entries associated with this keyword, and that each entry can be a mask that may contain the asterisk (*) and question mark (?) wildcard characters. A CNP user whose ID matches any mask in this list is authorized to issue a take action command. These CNP user IDs are defined within the CMS and do not necessarily exist on the system on which KWWAGENT is running.

Note: *You must include the TakeActionAuthUsers parameter in the KWWXML configuration file in order to enable take action commands. No one can issue take action commands unless this parameter is specified.*

Version

This keyword gives the version of the product for which the parameters were set. It is reserved for future use.

WASAppServerRoot

This keyword indicates the HFS directory path name to the location where the WebSphere Application Server for OS/390 is installed.

MonitorAppServer

The MonitorAppServer XML tag encapsulates keywords that apply to a single application server instance to be monitored by the agent. The MonitorAppServer tag is optional, and is only required if the monitoring requirements for a particular application server instance are different from the default settings for the agent. KWWAGENT automatically discovers each application server instance running in the domain of the WebSphere administrative server running on the same system. If there is no MonitorAppServer tag for an application server instance, then that server instance is monitored using the default KWWAGENT parameters. The following keywords apply to this tag. All are optional except the Name keyword.

CollectWorkloadClassName

Optional. This keyword is used in combination with the CollectWorkloadClassType keyword to specifically identify those classes for which workload analysis data is to be collected for this application server. In order to collect workload analysis data for a class, the class must have a type on the CollectWorkloadClassType list, and have a name that matches one of the CollectWorkloadClassName masks. The valid format is one or more class name masks separated by commas or semi-colons. Each class name mask may contain the asterisk (*) and question mark (?) wildcard characters.

CollectWorkloadClassType

Optional. This keyword specifies the type of workload analysis data that will be collected by default for all application servers. The valid format is an alphanumeric string of up to 10 characters. Each valid value, shown in the following list, represents a specific type of workload profiling data that should be collected.

Value	Meaning
A	Indicates that all workload data should be collected
B	Indicates that workload data for EJB-related classes should be collected
D	Indicates that workload data for SQL-related classes should be collected
I	Indicates the workload data of JNDI-related classes should be collected
M	Indicates that workload data for user-defined methods should be collected
N	Indicates that workload data for socket-related classes should be collected

Value	Meaning
Q	Indicates that workload data for JMS-related classes should be collected for Java
S	Indicates that workload data for servlet classes should be collected
T	Indicates that workload data for transaction-related classes should be collected

ExceptionWorkloadMax

Optional. This keyword sets a limit on the number of exception transactions for each workload identified by the ExceptionWorkloadName parameter. The valid format contains a positive integer that specifies the maximum number of exception transactions to collect.

If you issue a workload exception request for a workload mask that identifies multiple workloads, then the exception transactions for all the workloads that match the mask are collected. For example, if you have two workloads that match the mask, and you have this parameter set to 10, the agent would report a maximum of 10 exception transactions for each workload. The workload exception table view, in this case, would display up to 20 entries.

ExceptionWorkloadMinRespTime

Optional. This keyword specifies a response time threshold for reporting exception transactions. Any transactions having lower response times than the threshold value are not reported. The valid format contains an integer that specifies the number of milliseconds.

ExceptionWorkloadName

Optional. This keyword identifies the workloads running on this application server for which exception transactions are to be collected. It contains one or more masks that identify the class or classes of the workloads for which exception transactions are to be collected. Each mask may contain the asterisk (*) and question mark (?) wildcard characters. If this keyword is omitted, workload exception data is not collected for this application server unless data collection is specified for the KWWAGENT XML tag or by a SetException take action command.

Name

Required. This keyword specifies the name of the WebSphere Application Server for z/OS and OS/390 application server instance to which this tag applies.

Exclude="Yes|No"

Optional. This keyword indicates whether or not the application server instance should be monitored. If Exclude=Yes is specified, all other MonitorAppServer parameters, with the exception of Name are ignored. If Exclude=No is specified, then it overrides the DefaultExclude=Yes parameter which may have been specified at the agent level.

JVMProfilerCollect

Optional. This keyword specifies the type of JVM statistical data that will be collected. The value specified here is a character string of up to 5 characters. Each character represents one of the following specific types of JVM profiling data for collection:

Value	Meaning
C	Java monitor contention
H	Java heap usage
T	Thread statistics (implied if C or M options are set)
M	Java method invocation
G	Garbage collection

For example, if this keyword is set to a value of **CTM**, then the profiler will collect monitor contention, thread statistics, and method invocation data. If the JVMProfilerCollect parameter is set to an empty string, it indicates that JVM statistical information is not to be collected. An empty string is the default for the JVMProfilerCollect parameter.

JVMProfilerCollectStopAfter

Optional. This keyword specifies how long, in seconds, the collection of JVM statistics for an application server instance should continue after being enabled before it is automatically disabled. If this value is 0, it indicates that

JVM collection should continue until it is explicitly stopped. If this value is not zero, it should be greater than or equal to the value set for StatisticInterval.

JVMProfilerCollectClasses

Optional. This keyword indicates the Java classes for which the profiler will collect method, heap, and contention data for this application server instance. This keyword can be specified as a list of one or more names, separated by commas. Each entry can be a mask that can include the asterisk (*) and question mark (?) wildcard characters. Only Java class names that match one of the specified masks will have method, heap, and contention data collected by the profiler. If this parameter is not specified, then the value of the DefaultJVMProfilerCollectClasses parameter, specified at the KWWAGENT level, is used for this application server instance.

JVMProfilerCPUSampleInterval

Optional. This keyword indicates how often CPU sampling should be performed within the JVM for this application server instance. The specified value is in milliseconds. The smaller this value, the higher the overhead for JVM data collection, but the more accurate thread and method invocation data will be. The default is 5. This value is only meaningful if the DefaultJVMProfilerCollect keyword contains 'T' or 'M'.

JVMProfilerSampleInterval

This keyword provides the default sampling interval, in seconds, for profiler data for this application server. The KWWAGENT will collect profiler data from the application server instance based on the time specified. The default is 60.

JVMProfilerIgnoreMethods

Optional. This keyword indicates the names of the Java methods for which CPU busy samples are to be ignored by the profiler for this application server instance. On certain platforms, for example, AIX, the profiler interface incorrectly reports a "CPU busy" condition for certain Java methods. This can produce misleading results in the Java Threads and Java Methods reports. This keyword can be used to indicate that the CPU sample should ignore "CPU busy" conditions reported by the JVM profiler interface if the active method matches one of the entries in this parameter.

This keyword can be specified as a list of one or more names, separated by commas. Each entry can be a mask that can include the asterisk (*) and question mark (?) wildcard characters. CPU busy conditions reported against all Java method names that match one of the specified masks will be ignored. The default setting for this parameter is empty.

If this keyword is not specified, then the value of the DefaultJVMProfilerIgnoreMethods parameter, specified at the KWWAGENT level, is used for this application server instance.

RetainHistory

Optional. This keyword determines the length of time, in seconds, that SMF historical data not yet requested should be retained by the agent. The default is 3600 seconds (60 minutes). If history is being collected, this value should be greater that the historical collection interval. If history is not being collected, this value should be set to zero.

SMFCollectBeans

Optional. This keyword determines which beans will have detailed bean-level data collected for them. Bean data is contained in the SMF J2EE Container Interval records. The valid format is one or more masks separated by commas. Each mask may contain the asterisk (*) and question mark (?) wildcard characters to identify a group of matching bean names.

SMFCollectClasses

Optional. This keyword determines which classes will have detailed class-level data collected for them. Class data is contained in the SMF MOFW Container Interval records. The valid format is one or more masks separated by commas. Each mask may contain the asterisk (*) and question mark (?) wildcard characters to identify a group of matching classes.

SMFCollectMethods

Optional. This keyword determines which methods will have detailed method-level data collected for them. Methods data is contained in the SMF MOFW and J2EE Container Interval records. The valid format is one or more masks separated by commas. Each mask may contain the asterisk (*) and question mark (?) wildcard characters to identify a group of matching methods.

StatisticInterval

Optional. This keyword is reserved for future use.

SubnodeAliasName

Optional. This keyword lets you specify an alias name for this application server instance for the purpose of creating a Candle Technologies (CT) subnode name representing the server. KWWAGENT normally constructs a subnode name for the server with the following format:

> ***<application-server-name>*****:*****<node-name>*****:KWW*****x***

where

> <*application-server-name*> is the name of the application server instance name.
>
> <*node-name*> is the name of the system on which the application server instance is running.
>
> KWW*x* is either KWWM for a MOFW server or KWWJ for a J2EE server.

Because of a 32-character restriction for CT node names, however, the application server instance name or the node name or both, can be truncated. This can result in duplicate subnode names if more than one application server instance running on a system have names prefixed with the same set of characters. The SubnodeAliasName parameter allows you to specify an alias name for this server that will be used as the <application-server-name> portion of the subnode name. This lets you ensure that subnode names can be unique.

TakeActionAuthUsers

Optional. This keyword is a mask that indicates which CNP users are authorized to issue take action commands for this application server instance. There can be multiple entries associated with this parameter, The valid format is one or more masks separated by commas. Each mask may contain the asterisk (*) and question mark (?) wildcard characters.

A CNP user whose ID matches any mask in this list is authorized to issue a take action command. Note that the CNP user IDs are defined within the CMS, and do not necessarily exist on the system on which the KWWAGENT is running. You can also specify this parameter at the agent level. If specified at both the agent and application server instance level, the mask lists are effectively merged for the application server instance. A user is allowed to perform a take action on an application server instance if the user ID matches an entry in either the agent-level or application server instance level TakeActionAuthUsers list.

Note: *You must include the TakeActionAuthUsers parameter in the KWWXML configuration file in order to enable take action commands. No one can issue take action commands unless this parameter is specified.*

Your Agent's Environment Variables

What are the environment variables?

OMEGAMON XE for WebSphere Application Server for OS/390 uses key environment variables that affect the general CT and CandleNet Portal operations that support the OMEGAMON XE agents. These environment variables are set in the *rhilev.midlev.*RKANPAR(KWWENV) dataset, which is created during installation.

Only the CT_CMSLIST environment variable is required. You define this value during installation. The default values for the others should be satisfactory for most installations. You can, however, change their settings as needed by your system.

Definitions of the KWWAGENT environment variables

The following is a list of environment variables that you can use for the agent for OMEGAMON XE for WebSphere Application Server for OS/390.

CT_CMSLIST

Required. This variable names the host machine that is running the CMS to which you want to connect. This variable is prefixed by the protocol used to reach that host, such as, ip: or sna:. You can list multiple hosts to provide the ability to connect to a backup CMS.

KBB_RAS1

Optional. You can use this variable to turn on tracing. The possible values for this variable are the same as with all CT products. To trace KWWAGENT, use UNIT:KWW ALL. For example, set the variable to **UNIT:KWW ALL** to get all the trace data from the OMEGAMON XE for WebSphere agent.

KWW_AGENT_COMPONENT_TRACE

Optional. This variable indicates whether or not the internal tracing of requests to KWWAGENT should be enabled. This is an internal, wrap-around trace that contains the flow of requests and responses to and from the OMEGAMON XE for WebSphere agent. The default is YES.

KWW_INTERNAL_TRACE

Optional. This variable indicates whether internal flow tracing within KWWAGENT should be enabled. This is an internal, wrap-around trace that shows the flow of control through the functions of the OMEGAMON XE for WebSphere agent. These are the six possible values for this variable:

Value	Meaning
NO	Internal trace is not enabled.
YES	Internal trace is enabled for all functions.
MINOR	Internal trace is enabled for all minor functions.
PRIVATE	Internal trace is enabled for all functions classified as PRIVATE, RESTRICTED, or PUBLIC.
RESTRICTED	Internal trace is enabled for all functions classified as RESTRICTED or PUBLIC.
PUBLIC	Internal trace is enabled for all functions classified as PUBLIC.

The default is NO.

KWW_HEAP_DEBUG

Optional. This variable indicates whether or not the option for heap debugging should be enabled for KWWAGENT. The default is NO.

Activating the JVM Profiler

Background

The JVM profiler is a feature of OMEGAMON XE for WebSphere Application Server for OS/390 that obtains statistics directly from the Java Virtual Machine (JVM) in which the application server is running. Because this feature can incur high overhead, we recommend that it *not* be run all the time, particularly on production application servers. The profiler is intended to be used over limited amounts of time only, preferably on test or development application servers.

JVM statistics

OMEGAMON XE for WebSphere Application Server for OS/390 provides facilities for obtaining JVM statistical information from the JVMPI interface. You can monitor these optional JVM statistics for each server instance, server or system.

Use the JVM_EXTRA_OPTIONS environment variable to do this. This causes the JVM to load the profiling library at startup. You also need to ensure that the monitoring agent library, KWWPROF, is available to the JVM by copying this library into the proper run-time directory.

To activate the JVM profiler and collect the data., you need to complete the following steps.

Step 1: Change the IBM application server configuration

Modify the IBM WebSphere configuration for the application server or servers so that they automatically load this profiler library into the Java Virtual Machines in which the servers are running.

1. Use the IBM WebSphere System Management Enhanced User Interface (SMEUI) Administration application to modify the configuration for application server instance or instances for which you want to run the profiler. The first task is to use the SMEUI to define a new conversation.

2. Find the application server or servers you which to modify, and add an environmental variable for the application server instance called **JVM_EXTRA_OPTIONS**. The value of this environmental variable is **-Xrunkwwprof:port=*nnnnn***, where *nnnn* is the value of the JVMProfilingPort keyword in the KWWXML configuration parameters.
3. Add (or modify) an environmental variable called **JAVA_COMPILER**. The value of this environmental variable should be set to **NONE**.
4. Find an environmental variable named **LIBPATH** and add the following to the end of the LIBPATH value:

   :/CANDLEHOME/kww

   where CANDLEHOME is the Candle HFS directory.
5. Using the SMEUI, you must Validate, Commit, and Activate the conversation containing your change. This action will stop and restart the application server instance.

   Refer to the IBM manual *System Management User Interface* (SA22-7838) for more information about the SMEUI.

Step 2: Enable data collection in the OMEGAMON XE Agent

Use one of the following methods, to enable data collection by KWWAGENT.

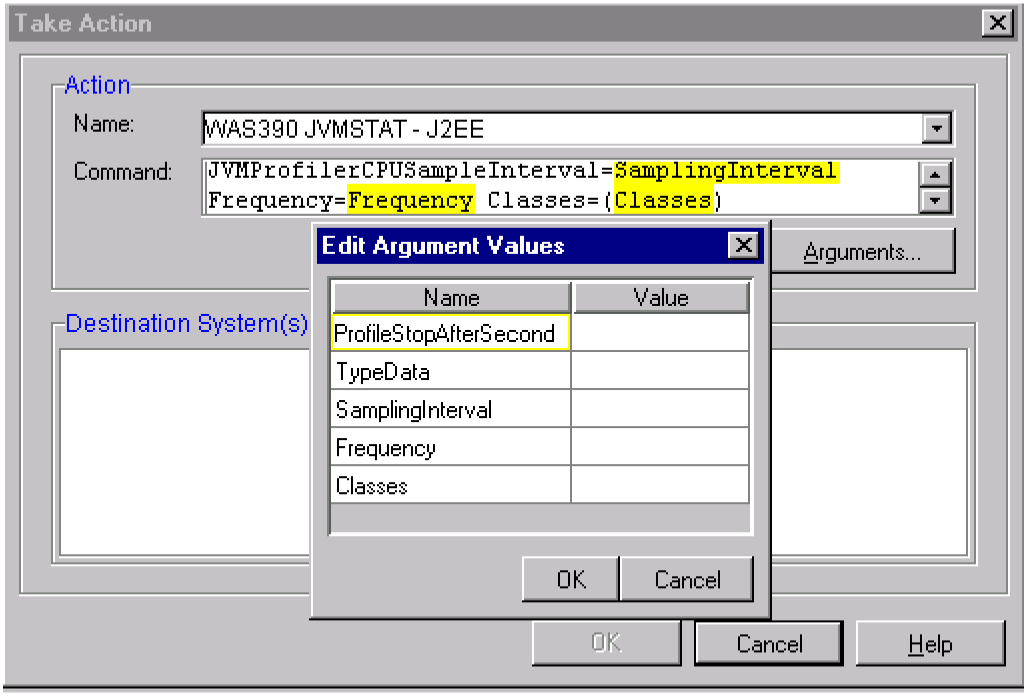
- Ensure that the DefaultJVMProfilerCollect parameter is set to collect the type of data you want. See "DefaultJVMProfilerCollect" on page 52.
- If the DefaultJVMProfilerCollect parameter is not set to collect data, use the WAS390 JVMSTAT - J2EE take action command to manage data collection for the JVM profiler.

  When you initiate this take action command, CandleNet Portal provides you with a Take Action dialog. Specify the arguments that you want to use for this command as explained in the following steps.

1. In the CandleNet Portal Navigator, select the application server for which you want to collect data. Right-click and choose **Take Action** from the pop-up menu.
2. In the Take Action dialog, scroll the names of the command in the Name field and choose **WAS390 JVMSTAT - J2EE**.

After you enable JVM data collection you will be able to view the data in the JVM workspaces and create situations to monitor it.

3. In the Take Action dialog, click the **Arguments** button to display the Edit Argument Values pop-up shown below.



Name: WAS390 JVMSTAT - J2EE

Command: JVMProfilerCPUSampleInterval=SamplingInterval Frequency=Frequency Classes=(Classes)

Name	Value
ProfileStopAfterSecond	
TypeData	
SamplingInterval	
Frequency	
Classes	

4. Complete the Value fields. The table below summarizes the information in these fields, which is detailed earlier in this chapter.

Table 2. JVMSTAT Take Action Command Values

Parameter Name	Value Explanation
ProfileStopAfter Second	Automatically stops the JVM profiler after the number of seconds you specify. See "DefaultJVMProfilerCollectStopAfter" on page 53.
TypeData	Determines the type of JVM statistical data to collect based upon the data type codes (C, H, T, M, and G) you enter. For example, CHTMG collects all data types. See "DefaultJVMProfilerCollectClasses" on page 53.

Table 2. JVMSTAT Take Action Command Values (continued)

Parameter Name	Value Explanation
SamplingInterval	Indicates how often CPU sampling should be performed within the JVM for this application server instance. The specified value is in milliseconds. See "DefaultJVMProfilerCPUSampleInterval" on page 53.
Frequency	Determines, in seconds, how often data is sent back to the agent. See "DefaultJVMProfilerSampleInterval" on page 54.
Classes	Determines which Java classes will be used for data collection. For example, * collects all classes. See "DefaultJVMProfilerCollectClasses" on page 53.

See "Your Agent's Configuration Parameters and XML Tags" on page 50 for more information about the values to use for each of these parameters.

Your parameter values depend upon the data and applications you want to monitor, the level of data you want to collect, and your overhead considerations.

Basics about Workload Analysis

Overview

OMEGAMON XE for WebSphere Application Server for OS/390 includes a workload analysis capability that enables you to collect and display workload data to see where your workloads are spending their time. These workloads are units of work the you define and track.

You can use this feature to see if workloads indicate efficient operations or if workloads consume capacity beyond the limits of the hardware and software being used. For example, workload analysis may indicate a problem with SQL calls to a database or higher resource usage than expected.

Background

After you initially install OMEGAMON XE for WebSphere Application Server for OS/390, you can enable an optional feature for workload data collection. To use this feature, perform the following procedures.

1. Define and set up your workloads in the kwwiuser.xml instrumentation file. The default location of this file is in the C:\CANDLEHOME\KWW folder.
2. Run the Candle class-loader setup utility that will enable workload analysis for the OMEGAMON XE agent.

   Note: *You will need to run this utility again, whenever you apply IBM maintenance* that affects the Java Development Kit (JDK).
3. Change the configuration for the IBM WebSphere Application Server for OS/390 product to enable the collection of workload analysis data.
4. Use the WAS390 Dynamic Workload Analysis take action command, or specify the appropriate Workload Analysis parameters in the KWWXML member, to control the collection of workload analysis data.

Moving on

Follow the procedures explained in the remainder of this chapter to setup workload data collection. Then, refer to "Workload Information for Performance Tuning" on page 157 for details about the workload analysis workspaces and managing this data collection. See the Help for OMEGAMON XE for WebSphere Application Server for OS/390 for details about using the WAS390 Dynamic Workload Analysis take action command and managing data collection for specific classes.

Setting Up the Workload Analysis Control File

Overview

The following topics provide a sample of the instrumentation file and the instrumentation settings that you use to implement this optional feature.

What is the default instrumentation file?

After you initially install OMEGAMON XE for WebSphere Application Server for OS/390, you will need to define instrumentation for your workloads. To do this, define the Java classes and methods that you want to track in the workload analysis control file called kwwiuser.xml.

You can find a Candle-provided sample file for kwwiuser.xml in rhilev.midlev.TKANSAM(KWWSAXML). This sample is shown in Figure 3 on page 73. In order to use this sample file, you need to copy it to the /CANDLEHOME/kww HFS directory, using the OPUT TSO command:

```
OPUT   'rhilev.midlev.TKANSAM(KWWSAXML)'
/CANDLEHOME/kww/kwwiuser.xml
```

You can then customize the resulting HFS file to provide the definitions for the proper instrumentation of your workloads.

For a detailed description of its XML tags and keywords, see the definitions that begin on page 73.

Sample kwwiuser.xml workload analysis control file

This is a sample of the kwwiuser.xml file that is installed on your system. Modify it to define the workloads that you want to monitor.

FIGURE 3. Default Workload Analysis Control File

```
<?xml version="1.0" encoding="UTF-8"?>
 <KWEINSTR Version="110"
                 MaxClasses="1000"
                 LogFileName="kwwinstr.log"
                 TraceActive="false"
                 CollectCPUTime="false"
                 SysInstr="Direct"
                 JVMProfilingPort="65535"
                 ShowInternalWorkloads="false"
 >
 <Class Name="Your.Class.Name"
         ClassType="user"
         MethodType="SERVLET,EJB">
 </Class>

 </KWEINSTR >
```

Keywords for the KWEINSTR tag

The KWWIUSER XML tag encapsulates all keywords for workload instrumentation. They provide overall controls for the instrumentation process. The definitions for each of these keywords are given in the order they appear in the file.

Version

This keyword identifies the version of the product for which the parameters were set. It ensures the parameter settings will be interpreted correctly in future releases. The current valid value and default values are 110.

MaxClasses

This keyword identifies the maximum number of classes to instrument. The valid format is a positive number; the default value is 1000.

CandleRoot

This keyword identifies the name of the Candle root directory. The valid format is a directory name. There is no default value; you must specify this.

SysInstr

This keyword indicates how instrumentation will be performed on the system classes. the maximum number of classes to instrument. The valid values are shown in the following table.

Value	Meaning
Direct	Indicates called system class methods will be instrumented
Indirect	Indicates the system class methods called from the user class will be instrumented

JVMProfilingPort

This keyword indicates the number to be used for communicating with the agent. It must match the JVMProfilingPort value that is in the kwwxml member. This parameter is only used when the JVM profiler is not running. IF the profiler is running, then the value of the “port” parameter in the -Xrun JVM runtime parameters is used. The valid format is a numeric port number. The default setting is 65535.

TraceActive

This keyword indicates whether or not the Workload Analysis component should write a detailed log file. The default value is false.

CollectCPUTime

This keyword indicates whether or not CPU usage data is to be collected. The valid values are true and false. The default value is false.

ShowInternalWorkloads

This keyword indicates whether or not workloads invoked by other workloads should be treated as separate workloads, or whether they should appear as delays to the original invoking workload. A value of 'true' indicates that internally invoked workloads are tracked as separate workloads. The default is false.

Keywords for the Class tag

The control file contains multiple Class entries. The Class XML tag specifies the type and level of instrumentation that is to be performed on the class or classes that match the class name parameter. The definitions for each of these keywords are given in the order they appear in the file.

Name

This keyword is a mask that identifies the names of the class or classes to which this statement applies. The valid format is one or more masks separated by commas or semi-colons. Each mask may contain the asterisk (*) and question mark (?) wildcard characters to identify a group of matching class names.

ClassType

This keyword indicates the type of class being identified by this statement. The valid values are shown below.

Value	Meaning
user	Indicates a user-defined class
system	Indicates a WebSphere system class

The default value is user.

MethodType

This keyword indicates the type of methods that should be instrumented with the class or classes. The valid values are shown below. There is no default value. Use commas to separate multiple values for this keyword.

Value	Meaning
EJB	Indicates that EJB-related methods should be instrumented
JMS	Indicates that JMS-related methods should be instrumented
JNDI	Indicates that JNDI-related methods should be instrumented
JTS	Indicates that methods relating to the Java Transaction Service (JTS) should be instrumented

Value	Meaning
METHODS	Indicates that method names explicitly specified by the MethodNames keyword are to be instrumented
SERVLET	Indicates that servlet-related methods should be instrumented
SOCKET	Indicates that socket-related methods should be instrumented
SQL	Indicates a WebSphere system class

The MethodType keyword must contain the METHODS value in order for the MethodNames keyword to be specified.

MethodNames

This keyword is a mask that identifies the names of the methods that are to be instrumented in addition to those identified by the MethodType parameter. The valid format is one or more masks separated by commas. Each mask may contain the asterisk (*) and question mark (?) wildcard characters to identify a group of matching method names.

SysInstr

This keyword indicates how instrumentation will be performed on the system classes. the maximum number of classes to instrument. The valid values are shown in the following table.

Value	Meaning
Direct	Indicates the called system class methods will be directly instrumented
Indirect	Indicates the system class methods from the user class will be indirectly instrumented

The default value is the value of SysInstr specified for the KWWINSTR parameter.

ActiveByDefault

This keyword indicates whether or not data collection for this class will be initially active. The valid values are true and false. The default value is false.

ScanForIndirect

This keyword indicates whether or not methods in this class should be scanned for calls to methods that are being indirectly instrumented. The valid values are true and false. The default value is true.

Configuring Your Agent for Workload Analysis

Overview

This topic explains how to configure the OMEGAMON XE for WebSphere agent for workload analysis after you initially install the product and define your workloads in the workload analysis control file. To complete workload configuration, you will

- run a shell script to enable workload analysis
- change the IBM WebSphere Application Server for OS/390 configuration

Background

After the initial install of OMEGAMON XE for WebSphere Application Server for OS/390, you should have an HFS directory (the default is CANDLEHOME) with a sub-directory called KWW. This sub-directory should contain the following files.

File	Contents
kwwwa.jar	This file contains all Candle-provided Java code for workload analysis.
kwwinstr.xml	This is the Candle-provided instrumentation control file. The contents of this file provide the basic instrumentation controls for the WebSphere Application Server provided Java classes. This class should not be edited directly. The contents of this file are logically merged with the contents of the kwwiuser.xml file when the Workload Analysis component is initialized. The instrumentation parameters in this file can be overridden in the kwwiuser.xml file. Edit this file to define your workloads as explained in“Setting Up the Workload Analysis Control File” on page 72.
Candle_WA_Enabler.class	This file contains the Candle extensions to the Java class loader environment that allows for dynamic instrumentation.
setupCandleWAEnabler.sh	This is a shell script that is used to invoke the Candle workload analysis enabler utility.

File	Contents
libkwwprof.so	This is a (binary) shared library that supports the Java profiler and workload analysis features.
kwwucoll	Binary used by agent for communicating with SMAPI

Run the shell script

Use the following steps to run the shell script that sets up the Candle workload analysis utility.

1. Use either Telnet or the OMVS TSO facility to obtain a UNIX System Services (USS) command prompt.
2. Enter

   ```
   cd /CANDLEHOME/KWW
   ./setupCandleWAEnabler.sh
   ```

   CANDLEHOME is the name of the HFS directory where the Candle products was installed.
3. Wait for the shell script to provide the following prompt.

   ```
   ============================================================
   Please enter the Java installation directory
   for your WebSphere Application Server 4.0.x runtime
   ============================================================
   ```
4. Provide an appropriate response for your site. For example,

   ```
   /WAS401/usr/lpp/java/IBM/J1.3
   ```
5. Wait for the shell script to provide the following prompt.

   ```
   ============================================================
   Please enter the USS directory for Candle KWW110 agent
   ============================================================
   ```
6. Provide the following reply.

   ```
   /CANDLEHOME/KWW
   ```

   CANDLEHOME is the name of the HFS directory where the Candle products was installed.
7. Wait for the shell script to create a new .jar file called **kwwscl.jar** in the /CANDLEHOME/KWW directory.

Modify the jvm.properties file

Modify the jvm.properities file after you run the shell script. Use the following procedures.

1. Locate the jvm.properties file in the following directory:

   CBCONFIG/controlinfo/envfile/SYSPLEX_NAME/SERVERINSTANCE_NAME/

   where CBCONFIG is the root directory in which WebSphere Application Server was installed.

2. Edit the jvm.properties file to add the following entry:

com.ibm.websphere.classloader.plugin=com.candle.kwe.instrument.ClassLoaderHook

Modify the jvm.properties file probably before making the additional configuration changes in the IBM WebSphere product described in the next topic.

Change the WebSphere Application Server for OS/390 configuration

Use the following steps to change the configuration for the WebSphere Application Server for z/OS and OS/390.

1. Use the IBM WebSphere System Management Enhanced User Interface (SMEUI) Administration application to modify the configuration for application server instance or instances. Use the (SMUI to define a new conversation.

2. Find the application server(s) you which to modify. You need to add three environmental variables for the application server instance.

   a.) The first environmental variable you need to add is JVM_EXTRA_OPTIONS. The value of this environmental variable depends on whether or not you want to run the Candle JVM profiler for this application server.

   - If you intend to run the JVM profiler, the value of JVM_EXTRA_OPTIONS should be **-Xrunkwwprof:port=*xxxxx*,workanalysisconfig=/CANDLEHOME /kww/kwwinstr.xml**

     where *xxxxx* is the value of the JVMProfilingPort, as found in the KWWXML file.

- If you do not want to run the JVM profiler, the value of JVM_EXTRA_OPTIONS should be **-Dcom.candle.kwe.instrument.control_file=/CANDLEHOME/kww/kwwinstr.xml**

In either case, CANDLEHOME is the name of the Candle HFS directory.

b.) The second environmental variable you need to add is JVM_BOOTCLASSPATH. The value of this environmental variable is

/CANDLEHOME/kww/kwwscl.jar:/CANDLEHOME/kww/kwwwa.jar

where CANDLEHOME is the name of the Candle HFS directory.

c.) The third environmental variable you need to add is JVM_BOOTLIBRARYPATH. The value of this environmental variable is

/CANDLEHOME/kww:/JAVAHOME/bin

where CANDLEHOME is the name of the Candle HFS directory, and JAVAHOME is the name of the JAVA directory.

d.) If you intend to run the Candle JVM profiler in this application server instance, add (or modify) an environmental variable called JAVA_COMPILER. The value of this environmental variable should be set to **NONE**.

e.) This step is only necessary if you intend to run the Candle JVM profiler. You should also find an environmental variable named LIBPATH. Add the following to the end of the LIBPATH value:

:/CANDLEHOME/kww

where CANDLEHOME is the Candle HFS directory.

3. Using the SMEUI, you must Validate, Commit, and Activate the conversation containing your change. This action will stop and restart the application server instance. Refer to the IBM manual *System Management User Interface* (SA22-7838) for more information about the SMEUI.

3

Introducing OMEGAMON XE for WebSphere Application Server for OS/390 Information

Overview

This chapter introduces you to the types of information that is reported by OMEGAMON XE for WebSphere Application Server for OS/390. It explains how to access the information and provides a summary of its organization.

Chapter contents

Data for WebSphere Application Server for OS/390

Introduction

OMEGAMON XE for WebSphere Application Server for OS/390, like many other Candle agents, provides data that helps you monitor the health and performance of your WebSphere Application Server for OS/390 environment.

CandleNet Portal (CNP) displays this information in named workspaces. Within a given workspace, this data can be presented in a tabular report format. CandleNet Portal refers to this format as a table view. Data provided to workspace can also be displayed in the form of charts, graphs, or other formats that you choose.

Using the OMEGAMON XE information

OMEGAMON XE for WebSphere Application Server for OS/390 provides the ability to view information about the WebSphere Application Server for OS/390 system that you are monitoring.

You can use this information to

- track the status of your WebSphere Application Server environment, including the Daemon server, System Management Server (SMS) server, IR server, Naming server, and one or more application servers instances
- monitor the performance of each WebSphere application server, helping you to evaluate tuning decisions
- monitor the performance of your J2EE and MOFW servers and their deployed applications
- use situations to report exceptions and choose the most effective thresholds for your needs
- review status information when a change in the state of a given resource occurs; such as from OK to Warning, or Warning to Critical
- perform bottleneck analysis for servlets, EJB methods, and Java Server Pages (JSP) by using workload analysis
- look at thread-related activity such as the number of threads and their CPU usage

- obtain heap-related data and determine garbage collection frequency
- use the application trace capability to follow the inter-method flows or control within the logic of your application
- track active HTTP sessions, see how often they are created and invalidated, and determine the number of objects that are bound to the sessions
- find out how often databases are being accessed, the average response time of database requests, and how long applications have to wait for connections from the database
- see how WebSphere applications are using MQSeries, find out which queue managers and queues are being used, and determine how many messages are being read and written

Associating table views and graphic views with attributes

There is a direct relationship between attribute groups in OMEGAMON XE for WebSphere Application Server for OS/390 and the table views and graphic views that display data in the workspaces. Individual attributes within a group correspond to the columns in the table views and data items presented by a chart shown in a workspace.

For example, the Application Server table view displays information collected by the Application Server attribute group. Each table view displays real-time information for many attributes. You can also display historical data collected over a time period that you define.

Data in the table views and graphic views

If there is no data available for a particular attribute, the corresponding column in a table view will be blank as well as any graphic presentations of the data. Because of high overhead, some data is not automatically reported in certain workspaces, for example, the Java Virtual Machine (JVM) data is turned off to avoid incurring the overhead associated with its collection. To collect this data and display it in the JVM workspaces, you need to use certain configuration settings and activate the Candle JVM profiler.

The configuration and specific workspace information in the other chapters explain the different requirements and procedures for collecting various data.

Accessing OMEGAMON XE Information

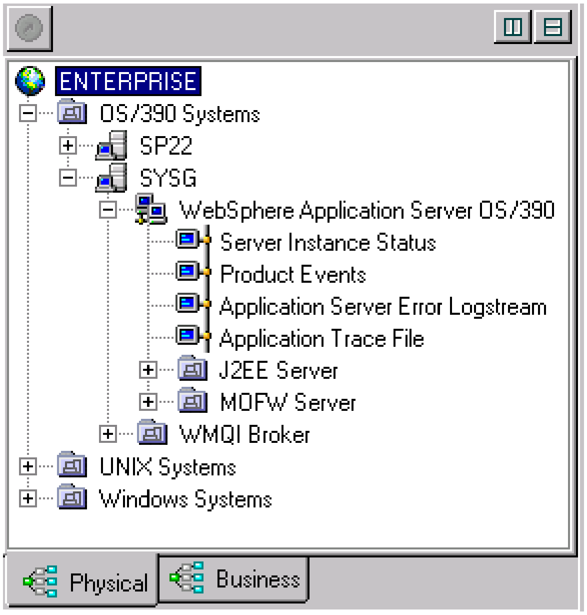
Accessing workspaces for OMEGAMON XE agent

To access the workspaces for OMEGAMON XE for WebSphere Application Server for OS/390, start from the Navigator. The hierarchy levels shown in the Navigator are dependent upon your enterprise. From the Navigator, do the following.

1. Choose **WebSphere Application Server OS/390** in the Navigator. This action expands the Navigator and opens the workspace for the agent, which shows overall status.

   If you are monitoring one server, the expanded Navigator reveals workspaces that are common to all servers and the J2EE Server and MOFW Server items shown below.

FIGURE 4. Navigator Example for OMEGAMON XE for WebSphere Application Server for OS/390

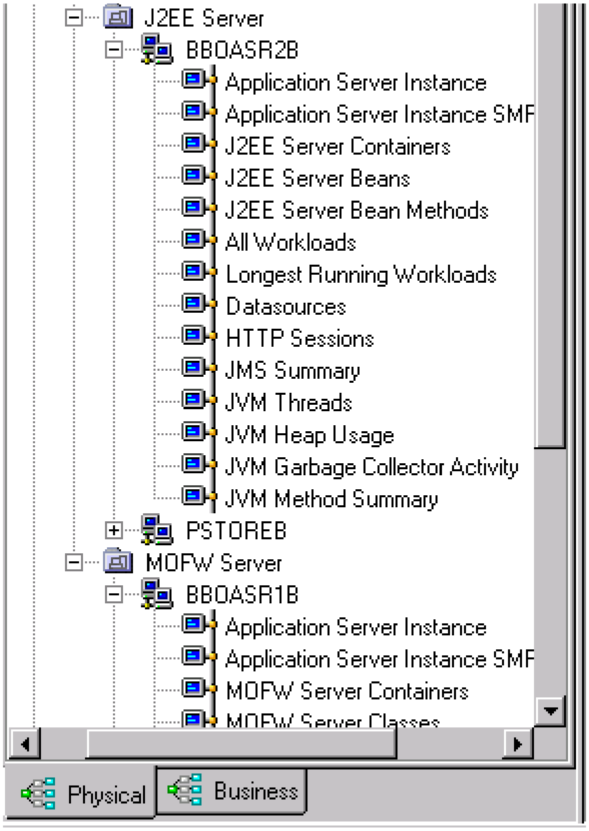


2. Expand the Navigator trees for the J2EE Server and MOFW Servers.

   If you are monitoring multiple application servers, they will be listed under the J2EE Server or MOFW Server items.

3. Expand the individual application servers in the Navigator to see the list of primary workspaces as shown below.

FIGURE 5. Expanded Navigator for J2EE and MOFW Servers



- J2EE Server
  - BBOASR2B
    - Application Server Instance
    - Application Server Instance SMF
    - J2EE Server Containers
    - J2EE Server Beans
    - J2EE Server Bean Methods
    - All Workloads
    - Longest Running Workloads
    - Datasources
    - HTTP Sessions
    - JMS Summary
    - JVM Threads
    - JVM Heap Usage
    - JVM Garbage Collector Activity
    - JVM Method Summary
  - PSTOREB
- MOFW Server
  - BBOASR1B
    - Application Server Instance
    - Application Server Instance SMF
    - MOFW Server Containers
    - MOFW Server Classes

4. Choose any item in the Navigator to view its associated workspace. (Grayed out items are not available for viewing.)

   You can also access additional workspaces, such as, history spaces from within a selected workspace.

Historical Reporting

Collecting short-term history

In addition to the long-term historical data collection reports, CandleNet Portal also provides a short term history reporting capability. You can find information on how to request short term history reports and how to specify the time interval for which you want short term history displayed in the Help for CandleNet Portal.

Default historical reports

Workspaces can provide additional views of historical data. You can specify a time span over which to collect historical data. The default setting collects data every 15 minutes. You can modify this setting to suit your needs. The OMEGAMON XE for WebSphere Application Server for OS/390 provides the following default historical views:

- Application Server Error Logstream History
- Application Server Instance SMF Interval Statistics History
- Longest Running Workload History
- J2EE Server Containers History
- MOFW Server Containers History
- Product Events History
- Selected Datasource History
- Selected Longest Running Workload History
- Selected Queue History
- Selected Workload Delay History
- Selected Workload History

Additional historical reports

In addition to the default historical reports, you can gather historical reporting data for all workspaces. Historical data can be displayed in the table, bar chart, pie chart, and plot chart views when historical collection has been configured and started for the Candle agent or agents associated with the view.

Use these steps to configure additional historical reports.

1. Open the workspace with the view for which you want historical data or define a historical workspace if none exists.
2. In the CandleNet Portal interface, choose History Configuration tool in the tool bar shown here or press Ctrl-H.
3. Complete the information for History Collection Configuration dialog box. See the online Help for CandleNet Portal for details.

   Note: *Before you can use the history feature, you must start historical data collection for the Candle agent or agents you want to report. See the Historical Data Collection Guide for OMEGAMON XE and Candle Command Center for details.*

4. To see the collected historical data, do the following:

   A. Choose the Time Span icon on the left, upper corner of a view.

   B. Choose the appropriate time your want to see.

   C. Check the **Apply to all views** associated with this view's query check box.

   D. Click **OK**.

OMEGAMON XE Workspaces

Overview

The workspaces for OMEGAMON XE for WebSphere Application Server for OS/390 are your windows to the data reported for WebSphere Application Server for OS/390. They provide you with status, definitions, and statistical information for the components of a WebSphere Application Server system.

Available workspaces

OMEGAMON XE for WebSphere Application Server for OS/390 provides the following workspaces.

System-wide information for WebSphere Application Server for OS/390

- *WebSphere Application Server OS/390*: This workspace displays enterprise-wide events and status of all server instances in your WebSphere environment.
- *Server Instance Status*: This workspace displays information about enterprise events as well as status and summary information for server instances on a specific system.
- *Product Events*: This workspace displays enterprise events and messages related to the operation the Candle OMEGAMON XE agent and the WebSphere Application Server for OS/390 on a specific system.
- *Application Server Error Logstream*: This workspace reports server exception and error conditions as recorded in the log files for server instances on a specific system.
- *Application Server Trace Files*: This workspace lists the trace files that you have created to track the inter-method flows or interaction of various components in your applications.
  - *Application Server Trace*: This workspace shows detailed information about the entries stored in a specific trace file.

Information about J2EE application servers

- *J2EE Server*: This workspace reports enterprise, status, and summary data for one or more J2EE servers on a system.
- *Server_name*: This workspace displays information about product events and status and configuration information for a specific J2EE server instance. The *server_name* variable is the name of the J2EE server instance in your enterprise.
- Application Server Instance: This workspace reports the enterprise, region size, and status and summary data for a specific J2EE server instance.
  - *Definition Environmental Variables*: This secondary workspace reports the names and values for definition environment variables for the application server instance.
  - *RunTime Environmental Variables*: This secondary workspace reports the names and values for runtime environment variables for the application server instance.
- *Application Server Instance SMF Interval Statistics:* This workspace displays aggregated information about all activities that occurred in a J2EE application server within the specified interval. The data is collected from the System Management Facility (SMF) server interval record.
- *J2EE Server Containers:* This workspace displays aggregated information about all activities that occurred in the container in a J2EE application server within the specified interval. The data is collected from the SMF container interval record.
- *J2EE Server Beans:* This workspace displays performance information for all active Enterprise Java beans (EJBs) in the container within the specified interval. The data is collected from the SMF container interval record.
- *J2EE Server Methods:* This workspace displays performance information for all active EJVB methods in the container within the specified interval. The data is collected from the SMF container interval record.
- *All Workloads:* This workspace reports summary information about all instrumented workloads and their response times.
  - Selected Workload - Current Interval: This workspace reports detailed information about specific response time.

  - Selected Workload - History: This workspace reports detailed workload information over an extended period response time.
  - Selected Workload Delays - Current Interval: This workspace reports detailed information about specific response times and CPU usage.
  - Selected Workload Delays - History: This workspace reports detailed information about specific response times and CPU usage over an extended period of time.
  - Ten Most Frequently Used Workloads: This workspace reports data about workloads that were most often executed during the interval.
  - Selected Longest Running Workload - History: This workspace reports detailed information about delays over an extended period of time.
- *Longest Running Workloads:* This workspace reports information about the individual transactions that exceeded your response time limits.
- *Datasources:* This workspace reports such information as connection, query, and update data for configured data sources.
  - *Selected Datasource - History:* This workspace reports connection, query, and update data for a selected data source over an extended period of time.
- *HTTP Sessions:* This workspace reports data related to sessions that area maintained by the application servers.
- *JMS Summary:* This workspace reports data related to the use of MQSeries by J2EE applications.
  - Selected Queue History: This workspace reports information about a specific MQSeries queue over an extended period.
- *JVM Threads*: This workspace displays information about specific threads running in a JVM and graphic views of CPU usage and threads waiting o enter the monitor.
  - *JVM Methods*: This secondary workspace displays information about specific methods running in the JVM and graphic views of CPU usage and methods waiting to enter the monitor.
  - *JVM Monitor Contention*: This secondary workspace displays information about monitor contention and a graphic view of threads waiting to enter the monitor.

- *JVM Heap Usage*: This workspace displays information about heap usage and graphic views of heap size and heap allocation rates.
- *JVM Garbage Collector Activity*: This workspace displays information about the JVM's garbage collector and graphic views of garbage collection activities.
- *JVM Method Summary:* This workspace displays information about methods being invoked in the JVMs that are hosting application servers.

Information about MOFW servers

- *MOFW Server*: This workspace reports enterprise, status, and summary data for one or more managed object framework (MOFW) servers on a system.
- *Server_name*: This workspace displays information about product events, application server error logstream information, status, and configuration information for a specific MOFW server instance on a system. The *server_name* variable is the name of the MOFW server instance in your enterprise.
- Application Server Instance: This workspace reports the enterprise, region size and status, and summary data for a specific MOFW server instance.
  - *Definition Environmental Variables*: This secondary workspace reports the names and values for definition environment variables for the application server instance.
  - *RunTime Environmental Variables*: This secondary workspace reports the names and values for runtime environment variables for the application server instance.
- *Application Server Instance SMF Interval Statistics*: This workspace displays aggregated information about all activities that occurred in a MOFW application server within the specified interval. The data is collected from the SMF server interval record.
- *MOFW Server Containers*: This workspace displays aggregated information about all activities that occurred in the container in a MOFW application server with the specified interval. The data is collected from the SMF server interval record.

- *MOFW Server Classes*: This workspace displays aggregated information about all active methods in the container within the specified interval. The data is collected from the SMF server interval record.
- *MOFW Server Methods*: This workspace displays aggregated information about all active methods in the container within the specified interval. The data is collected from the SMF server interval record.
- *JVM Threads*: This workspace displays information about specific threads running in a JVM and graphic views of CPU usage and threads waiting o enter the monitor.
  - *JVM Methods*: This secondary workspace displays information about specific methods running in the JVM and graphic views of CPU usage and methods waiting to enter the monitor.
  - *JVM Monitor Contention*: This secondary workspace displays information about monitor contention and a graphic view of threads waiting to enter the monitor.
- *JVM Heap Usage*: This workspace displays information about heap usage and graphic views of heap size and heap allocation rates.
- *JVM Garbage Collector Activity*: This workspace displays information about the JVM's garbage collection and graphic views of garbage collection activities.

Each of these workspaces is covered in greater detail in the remainder of this book.

4

Monitoring Enterprise Status and the OMEGAMON XE Agent

Overview

This chapter discussed the workspaces that you can use to monitor the overall status of a WebSphere Application Server for OS/390 environment as well as the status of the deployed OMEGAMON XE for WebSphere Application Server for OS/390 agent.

Chapter contents

Introduction

Views of high-level information

This chapter discusses workspaces that provide high-level information for your administrative needs. These workspaces report current status and availability of WebSphere Application Server for OS/390 and the KWWAGENTs for OMEGAMON XE for WebSphere Application Server for OS/390 you have installed on your system.

The workspaces that provide this high-level information are the

- WebSphere Application Server OS/390
- Server Instance Status
- Product Events
- Application Server Error Logstream

Organization of information

The topics in this chapter discuss each of these high-level workspaces. Each topic summarizes the purpose of the workspace and describes its contents.

Introducing the WebSphere Application Server OS/390 Workspace

Overview

The WebSphere Application Server OS/390 workspace displays information about important events in your enterprise. These events include situations you are monitoring for your WebSphere Application Server. It also reports the status and availability of WebSphere application server instances that are being monitored by an active OMEGAMON XE agent for WebSphere Application Server for OS/390.

Accessing the WebSphere Application Server OS/390 workspace

To access the workspace for the WebSphere Application Server OS/390 workspace, do the following:

1. Expand the Navigator.
2. Choose the WebSphere Application Server OS/390 item.

Inside the WebSphere Application Server OS/390 workspace

The default WebSphere Application Server OS/390 workspace contains

- an Product Events table view that displays events issued by the OMEGAMON agents
- an Application Server Error Log table view
- a WebSphere Application Server Summary table view

Use

The Application Server Error Log table view displays details exception and error messages from the error logstream of your WebSphere application server. For details, see the "Information presented in the Application Server Error Log table view" on page 104. The WebSphere Application Server Summary table view provides you with the status of each server instance that is being monitored by an active agent. If the agent is not active, then these instances are not listed in the table view. Server instances that are not being monitored by an active agent will also be grayed out in the Navigator.

Information presented in the WebSphere Application Server Summary table view

In addition to status, the table view also includes key information such as server type, release level, and Lightweight Directory Access Protocol (LDAP) status and port. The following is a list of the Server Instance Status attributes that can be reported in this table view.

- Build Level
- Control Region PROC Name
- Host Name
- LDAP IP Name
- LDAP Port Number
- LDAP Status
- Origin Node
- Release Level
- Server Instance Name
- Server Name
- Server Region PROC Name
- Server Role
- Server Type
- Status

For detailed information about these attributes and their values, refer to the online Help for OMEGAMON XE for WebSphere Application Server for OS/390.

Introducing the Server Instance Status Workspace

Overview

The Server Instance Status workspace reports significant events in your enterprise that affect all systems running Candle products with a view of the Enterprise Event Console. It also monitors product events and reports those that may affect performance data collection.

Accessing the Server Instance Status workspace

To access the Server Instance Status workspace,

1. Expand the Navigator.
2. Expand the WebSphere Application Server OS/390 node to access its nested items.
3. Choose the Server Instance Status selection in the Navigator.

Inside the Server instance Status workspace

The default Server Instance Status workspace contains an Enterprise Event Console and a Server Instance Status table view. The table view presents the status for all WebSphere server instances on a system.

Use

The Server Instance Status table view consolidates server instance status information in a single convenient table view. You can use this view to determine server type, role, release, and build levels along with other key information.

Information presented in the Server Instance Status table view

The information in this table view is reported by the Server Instance Status attributes. The following is a list of the attributes that can be reported in this table view.

- Build Level
- Control Region PROC Name
- Host Name
- LDAP IP Name
- LDAP Port Number
- LDAP Status
- Origin Node
- Release Level
- Server Instance Name
- Server Name
- Server Region PROC Name
- Server Role
- Server Type
- Status

For detailed information about these attributes and their values, refer to the online Help for OMEGAMON XE for WebSphere Application Server for OS/390.

Introducing the Product Events Workspace

Overview

The Product Events workspace reports significant events in your enterprise that affect all systems running Candle products with a view of the Enterprise Event Console. It also monitors the status of the OMEGAMON XE for WebSphere Application Server for OS/390 agent itself and reports those that may affect performance data collection.

Accessing the Product Events workspace

To access the Product Events workspace,

1. Expand the Navigator.
2. Expand the WebSphere Application Server OS/390 node to access its nested items.
3. Choose the Product Events selection in the Navigator.

Inside the Product Events workspace

The default Product Events workspace contains an Enterprise Event Console and a Product Events table view. The table view reports agent-level events that may affect the ability of the agent to collect data for the WebSphere Application Server OS/390.

Historical data

If you want to view this data over a particular time span, you can navigate to the Products Events History workspace or invoke the Time Span dialog to set a specific period of time. This history workspace also contains an Enterprise Event Console and a Product Events table view.

This historical data is collected over a measured time span. See the online Help for CandleNet Portal for detailed explanation of historical reporting.

To access the history workspace, use any of the following procedures.

- **Select View > Workspace** from the menu bar in the Product Events workspace and choose the history workspace from a pull-down menu.

- Select and right-click the Product Events item in the Navigator choose the history workspace from the pop-up menu.
- Click the link icon for a row in the Product Events table view to link to the history workspace.

Use

The Product Events table view displays error the events issued by the OMEGAMON XE agents. You can use this convenient view to see exception and error messages, their IDs, and severity. Place your cursor over a truncated message to display the text of the complete message.

Information presented in the Product Events table view

The information in this table view is reported by the Product Events for WAS OS/390 attributes. The following is a list of these attributes that can be reported in this table view.

- Event Date and Time
- File Name
- Function
- Host Name
- Line Number
- Message Description
- Message ID
- Origin Node
- Severity

For detailed information about these attributes and their values, refer to the online Help for OMEGAMON XE for WebSphere Application Server for OS/390.

Introducing the Application Server Error Logstream Workspace

Overview

The Application Server Error Logstream workspace reports server error and exception conditions as written to the error logstream for all WebSphere server instances. The workspace presents error and exception information for all the applications server instances being monitored by the OMEGAMON XE agent for WebSphere.

Accessing the Application Server Error Logstream workspace

To access the Application Server Error Logstream workspace for a system,

1. Expand the Navigator.
2. Expand the WebSphere Application Server OS/390 node to access its nested items.
3. Choose the Application Server Error Logstream selection in the Navigator.

Inside the Application Server Error Logstream workspace

The default Application Server Error Logstream workspace contains an Enterprise Event Console and an Application Server Error Log table view. The Application Server Error Log table view presents error and exception information for all WebSphere server instances defined on a system.

Historical data

If you want to view this data over a particular time span, you can navigate to the Application Server Error Logstream History workspace or invoke the Time Span dialog to set a specific period of time. This history workspace also contains an Enterprise Event Console and a Application Server Error Log table view.

This historical data is collected over a measured time span. See the online Help for CandleNet Portal for detailed explanation of historical reporting.

Information presented in the Application Server Error Log table view

The Application Server Error Log table view reports messages recorded in the logstream. The information displayed in this workspace includes the message ID and text as well as the message origin and other identifying information that will help you trace its source.

The information in this table view is reported by the Application Server Error Log attribute group. The following is a list of attributes in the group.

- Error Date and Time
- Host Name
- Job Address Space Identifier
- Job Name
- Message Identifier
- Message Origin
- Message Text
- Origin Node
- Process Identifier
- Server Instance Name
- Thread Identifier

For detailed information about these attributes and their values, refer to the online Help for OMEGAMON XE for WebSphere Application Server for OS/390.

5

J2EE and MOFW Servers

Overview

This chapter provides instructions for accessing workspaces available for J2EE and MOFW servers and the application servers they support. It also provides information about the contents of these workspaces.

Chapter contents

Introducing the Workspaces for J2EE and MOFW Servers

Introduction

The J2EE Server and MOFW Server workspaces report status and configuration information for each J2EE server instance or each MOFW server instance.

If there are multiple J2EE and MOFW servers in your WebSphere environment, they are listed in the CandleNet Portal Navigator as multiple systems underneath the J2EE Server or MOFW Server item. The names of these server instances are shown on the Navigator tree to represent each of them.

If there is only one J2EE or MOFW server in your installation, the workspaces that pertain to a single application server are listed under the J2EE or MOFW item in the Navigator.

Accessing the J2EE Server or MOFW Server workspaces

Use these steps to access the J2EE Server and MOFW Server workspaces.

1. Expand the Navigator.
2. Expand the WebSphere Application Server OS/390 node to access its nested items.
3. Choose the J2EE Server or MOFW Server selection in the Navigator.

Inside the J2EE Server or MOFW Server workspaces

The predefined J2EE Server and MOFW Server workspaces contain

- an Enterprise Event Console that displays changes in the status of situations that have been assigned to the console view
- a J2EE Servers Summary table view or a MOFW Servers Summary table view, depending on your workspace selection

J2EE Servers Summary and MOFW Servers Summary Table Views

Use

The J2EE Servers Summary and MOFW Servers Summary table views provide display status and configuration information for all the J2EE or MOFW servers that are being monitored by the OMEGAMON XE agent during the interval. Use these table views to review status and configuration information for each server instance in your WebSphere environment. For example, you can review information about stack sizes, policies, garbage collection, and ports.

Information presented in the J2EE Servers Summary and MOFW Servers Summary table views

The J2EE Servers Summary and MOFW Servers Summary table views display data provided by the Application Server Instance attributes. A list of these attributes appears on the next page.

These following attributes report the data in the J2EE Servers and MOFW Servers workspaces

- Allow Accept Asserted Identities
- Allow DCE
- Allow Kerberos
- Allow Non-Authenticated Clients
- Allow Send Asserted Identities
- Allow Server Region Garbage Collection
- Allow SSL Basic Authentication
- Allow SSL Client Certificates
- Allow Userid Passticket
- Allow Userid Password
- Configured Port Number
- Control Region Identity
- DCE Keytab File
- DCE Quality of Protection
- Debugger Allowed
- Garbage Collection Interval
- Host Name
- Isolation Policy
- Local Identity
- Logstream Name
- Number of HTTP Sessions
- Number of Regions
- OLT Host Name
- OLT Port
- Origin Node
- Production Server
- Region Started Rate
- Remote Identity
- Replication Policy
- Sample Date and Time
- Security Preference List
- Server Instance Description
- Server Instance Name
- Server Name
- Server Region Identity
- Server Region Stack Size
- SMF Container Activity
- SMF Container Interval
- SMF Interval Length
- SMF Server Activity
- SMF Server Interval
- SSL Firewall Port
- SSL RACF Keyring
- SSL Version 2 Timeout
- SSL Version 3 Timeout
- Status
- Transaction Factory

For detailed information about these attributes and their values, refer to the online Help for OMEGAMON XE for WebSphere Application Server for OS/390.

Introducing an Application Server Instance Workspace

Introduction

If you have multiple J2EE server instances running or MOFW server instances in your WebSphere environment, the Navigator lists these application servers by their names under the J2EE Server or MOFFW Server node. Each of these workspaces reports the status and summary data for a specific server instance.

Accessing an application server instance workspace

Use these steps to access an application server instance workspace.

1. Expand the Navigator.
2. Expand the WebSphere Application Server OS/390 node to access its nested items.
3. Select the J2EE or MOFW server instance you want to inspect and expand its Navigator tree.
4. Choose the name of application server you want to examine.

Inside an application server instance workspace

The predefined workspace for an application server instance contains

- an Enterprise Event Console that displays changes in the status of situations that have been assigned to the console view
- an Application Server Error Log that lists log entries for the application server instance
- an Application Server Instance table view that displays detailed information about the application server instance

Use

Use the Application Server Instance table view to review the status and configuration information for a specific server instance. For example, it reports status to let you know whether or not the instance is running or not and also many configured properties.

The Application Server Error Log table view displays detailed error and exception messages that were logged to the error logstream of the application server instance. For example, it provides the origin or the message, its ID, and the text of the message.

The Application Server Instance table view displays a wide-range of highly detailed data about the application server. For example, it provides security, access, policy, and SMF information about the server.

Information presented in the Application Server Error Log table view

The data in the Application Server Error Log table view is provided by the Application Server Error Log attributes. See "Application Server Error Log Table View" on page 204 for a listing of these attributes.

Information presented in the Application Server Instance table view

The data in the Application Server Instance table view is provided by the Application Server Instance attributes. See "Information presented in the J2EE Servers Summary and MOFW Servers Summary table views" on page 107 for a listing of the Application Server Instance attributes.

6

Application Server Instance and Environmental Variables Workspaces

Overview

This chapter provides instructions for accessing the Application Server Instance workspace and the associated Environmental Variables workspaces. It also discusses their contents.

Chapter contents

Introducing the Application Server Instance Workspace

Introduction

The Application Server Instance workspace reports status and configuration information about a specific server instance.

Accessing the Application Server Instance workspace

Use these steps to access the Application Server Instance workspace.

1. Expand the Navigator.
2. Expand the WebSphere Application Server OS/390 node to access its nested items.
3. Select either J2EE Server or MOFW Server and expand the Navigator tree.
4. Select the application server instance you want to inspect and expand its Navigator tree.
5. Choose the Application Server Instance selection in the Navigator.

Inside the Application Server Instance workspace

The predefined Application Server Instance workspace contains

- an Enterprise Event Console that displays changes in the status of situations that have been assigned to the console view.
- an Application Server Error Log table view that displays details about exception and error messages
- an Application Server Instance table view that displays general server information, such as, name, type, and release level for an application server instance

Application Server Instance Table View

Use

The Application Server Instance table view presents status and configuration information for a specific server instance. This table view also provides access to environmental variables. From the table view, you can select a row to link to the Definitional Environmental Variables and the Run Time Environmental Variables workspaces. Use it for quick access to summary information for general administrative purposes.

SMF information

Use this table view to determine if SMF data of server and container activity and server and container interval records are being gathered.

Information presented in the Application Server Instance table view

The Application Server Instance table view displays data provided by the Application Server Instance attributes. A list of these attributes appears on the next page.

The following attributes report the data in the Application Server Instance workspace.

- Allow Accept Asserted Identities
- Allow DCE
- Allow Kerberos
- Allow Non-Authenticated Clients
- Allow Send Asserted Identities
- Allow Server Region Garbage Collection
- Allow SSL Basic Authentication
- Allow SSL Client Certificates
- Allow Userid Passticket
- Allow Userid Password
- Configured Port Number
- Control Region Identity
- DCE Keytab File
- DCE Quality of Protection
- Debugger Allowed
- Garbage Collection Interval
- Host Name
- Isolation Policy
- Local Identity
- Logstream Name
- Number of HTTP Session Attributes
- Number of HTTP Sessions
- Number of Regions
- OLT Host Name
- OLT Port
- Origin Node
- Production Server
- Region Started Rate
- Remote Identity
- Replication Policy
- Sample Date and Time
- Security Preference List
- Server Instance Description
- Server Instance Name
- Server Name
- Server Region Identity
- Server Region Stack Size
- SMF Container Activity
- SMF Container Interval
- SMF Interval Length
- SMF Server Activity
- SMF Server Interval
- SSL Firewall Port
- SSL RACF Keyring
- SSL Version 2 Timeout
- SSL Version 3 Timeout
- Status
- Transaction Factory

For detailed information about these attributes and their values, refer to the online Help for OMEGAMON XE for WebSphere Application Server for OS/390.

Introducing the Environmental Variables Workspaces

Introduction

The environmental variables workspaces report the environment variables report the names and values of the environmental variables for the server instance. You can access these variables through the Definition Environmental Variables and RunTime Environmental Variables workspaces.

You can access these supplementary workspaces for an Application Server Instance by using any of the following methods.

Method 1: Accessing the environmental variables workspaces from the Application Server Instance table view

Use these steps to access the environmental variables workspaces from a table view.

1. Expand the Navigator.
2. Expand the WebSphere Application Server OS/390 node to access its nested items.
3. Select either J2EE Server or MOFW Server and expand the Navigator tree.
4. Select the application server you want to inspect and expand its Navigator tree.
5. Choose the Application Server Instance selection in the Navigator.
6. Select one of the following access paths to view the workspace you want.
   - To view the Definitional Environment Variables workspace, simply click the link icon in the table view beside the row of the server instance you want to examine.
   - To view the Run Time Variables workspace, right-click the link icon in the table view beside the row of the server instance you want to examine and then select RunTime Environmental Variables from the pop-up menu.

Method 2: Accessing the environmental variables workspaces from the Navigator

Use these steps to access the environmental variables workspaces from the navigator.

1. Expand the Navigator.
2. Expand the WebSphere Application Server OS/390 node to access its nested items.
3. Select either J2EE Server or MOFW Server and expand the Navigator tree.
4. Select the application server instance you want to inspect and expand its Navigator tree.
5. Right-click the Application Server Instance selection in the Navigator.
6. Choose Workspace from the pop-up menu and then either the Definitional Environmental Variables or RunTime Environmental Variables option.

Method 3: Accessing the environmental variables workspaces from the menu

Use these steps to access the environmental variables workspaces from the menu.

1. Expand the Navigator.
2. Expand the WebSphere Application Server OS/390 node to access its nested items.
3. Select either J2EE Server or MOFW Server and expand its Navigator tree.
4. Select the application server instance you want to inspect and expand its Navigator tree.
5. Choose the Application Server Instance selection in the Navigator to open the workspace.
6. From the menu, select **View > Workspace** and choose either the Definitional Environmental Variables or RunTime Environmental Variables from the pop-up menu.

Inside the Environmental Variables workspaces

The predefined Definition Environmental Variables and RunTime Environmental Variables workspaces contain

- an Enterprise Event Console that displays changes in the status of situations that have been assigned to the console view.
- a Definition Environmental Variables table view or RunTime Environmental Variables table view, depending on your choice of workspaces

From these workspaces, you can also access the Application Server Instance workspace and the other environmental variables workspace.

Environmental Variables Table Views

Use

The Definition Environmental Variables and RunTime Environmental Variables table views present the names and values for the defined environmental variables for a specific server instance. Use these environmental variables for a quick view of the runtime profile of the control region and the server region or regions.

Note: *Environmental data for RunTime Environmental Variable table view is obtained from the master console. If the agent starts later than the server instance, this table view is always empty.*

Information presented in the environmental variables table views

The environmental variables table views display data provided by the Environmental Variables attributes. The following is a list of these attributes.

- Host Name
- Parameter Name
- Parameter Value
- Server Name
- Server Instance Name
- Variable Type

For detailed information about these attributes and their values, refer to the online Help for OMEGAMON XE for WebSphere Application Server for OS/390.

7

System Management Facility Information

Overview

This chapter provides information about WebSphere for z/OS performance statistics that are produced by the Systems Management Facility (SMF) for WebSphere Application Server. It also discusses the workspaces that report this data and provides instructions for accessing them.

Background Information

Chapter goals

This chapter introduces you to the SMF information that is available in OMEGAMON XE for WebSphere Application Server. It identifies configuration parameters will use and discusses the workspaces in which you can view SMF information.

Chapter navigator

The chart below shows where you can locate information in this chapter about specific tasks you want to perform.

If you want to...	See...
identify workspaces that display detailed SMF data	page 121
enable the recording of SMF data in WebSphere Application Server for OS/390	page 123
identify which configuration parameters are used to collect SMF data	page 122
review communication and traffic information presented in the Application Server Instance SMF Statistics workspace	page 126
review information about active J2EE containers presented in the J2EE Server Containers workspace	page 130
review configuration information about deployed Java beans presented in the J2EE Server Beans workspace	page 131
review statistical information about Java bean methods presented in the J2EE Server Bean Methods workspace	page 133
review information about active containers presented in the MOFW Server Containers workspace	page 136
review the information presented in the MOFW Server Classes workspace	page 138
review the information presented in the MOFW Server Methods workspace	page 140

About SMF Information

Benefits

You can use SMF recording and monitoring to gather performance information from WebSphere Application Server for OS/390. Then, you can use this detailed information for purposes such as, monitoring resource usage, planning capacity, or billing. The OMEGAMON XE agent for WebSphere Application Server for OS/390 gives you fast access to this data by collecting and reporting the data from the server interval and container interval SMF records.

Monitoring SMF information with the OMEGAMON XE agent

OMEGAMON XE for WebSphere Application Server for OS/390 contains attributes and workspaces that report SMF data for both J2EE and managed-object framework (MOFW) servers. The workspaces that give you a direct view of this SMF data are the

- Application Server Instance SMF Interval Statistics
- Application Server Instance SMF Interval Statistics History
- J2EE Server Containers
- J2EE Server Containers History
- J2EE Server Beans
- J2EE Server Bean Methods
- MOFW Server Containers
- MOFW Server Containers History
- MOFW Server Classes
- MOFW Server Methods

These workspaces and the attributes that support them are described later in this chapter.

Note: *SMF recording must be enabled for the OMEGAMON XE agent to report data in these workspaces.*

Collecting the SMF Information You Want

Overview

To collect the type of SMF data you want, you need to manage both your IBM WebSphere Application Server for OS/390 settings for collecting SMF information and the OMEGAMON XE WebSphere agent parameters that control its reporting of SMF data. This topic supplies general information about how you set up this data collection process.

OS/390 requirements

To collect SMF data, you must have certain exit points active in OS/390. Enable SMF record 120 collection within SYS1.PARMLIB(SMFPRMxx) with the EXITS keyword by adding the following entry to the SMFPRM*xx* member:

```
TYPE(...120..)
EXITS(IEFU83, IEFU84, IEFU85)
```

IBM WebSphere Application Server settings

You need to enable SMF recording in WebSphere Application Server for OS/390 before the OMEGAMON XE agent can report this information. To do this, follow these steps.

1. Select the Server Definition or J2EE Server Definition option for the IBM System Management User Interface Administration application.
2. For the Server Definition or J2EE Server Definition, choose one of the following:
   - To collect server interval statistics, choose one or both of these checkboxes:
     - **Write Server Interval SMF Records**
     - **Write Container Interval**.

     Use this choice to enable data reporting in the Application Server SMF Interval Statistics workspaces.

- To collect container interval statistics, check the checkbox for **Write Container Interval SMF Records**.

  Use this choice to enable data reporting in these workspaces:

  – J2EE Server Containers
  – J2EE Server Beans
  – J2EE Server Bean Methods
  – MOFW Server Containers
  – MOFW Server Classes
  – MOFW Server Methods

3. Set the option for **SMF Interval Length**.

   If you specify a length of 0, the interval specified in the SMF parmlib member from the SMF product settings will take effect. This setting determines how often records are written and, therefore, controls the frequency at which the OMEGAMON XE agent will collect and update the SMF data in the table views.

   For additional information about SMF data collection, refer to the IBM publications, *System Management User Interface*, SA22-7838 *Operations* and *Administration*, “Setting up SMF recording,” SA22-7835.

   Note: *Ensure that you are not collecting more SMF data than you need and that the SMF dataset control interval (CI) sizes are large enough to ensure the most efficient writing of the SMF data to the dataset. Also ensure that the SMF datasets are allocated optimally.*

Configuration parameters for the OMEGAMON XE agent

Use configuration settings in the KWWXML member to manage collection of the SMF data. The settings that you use for the agent parameters and application server parameters in this member will tailor the collection of SMF data to meet your needs. Rather that just delivering a quantity of general SMF data, these parameters enable you to gather the specific data in which you are interested.

The following table provides a summary of these configuration parameters. For detailed information about using each of these parameters and the

KWWXML member, see Chapter 2, "Configuring OMEGAMON XE for WebSphere Application Server for OS/390" on page 39.

Table 3. Summary of SMF Configuration Parameters

Parameter	Function
Agent Parameters	
DiscardSMFRecords	Determines whether or not the processed SMF records will be discarded before being written to the SMF dataset
DefaultSMFCollectClasses	Identifies MOFW classes data to collect from SMF records
DefaultSMFCollectBeans	Identifies J2EE bean data to collect from SMF records
DefaultSMFCollectMethods	Identifies MOFW and J2EE method data to collect from SMF records
DefaultRetainHistory	Identifies how long the agent will retain historical data that it will collect in the future
Application Server Parameters	
SMFCollectClasses	Identifies MOFW classes data to collect from SMF records
SMFCollectBeans	Identifies J2EE bean data to collect from SMF records
SMFCollectMethods	Identifies MOFW and J2EE method data to collect from SMF records
RetainHistory	Identifies how long the agent will retain historical data that it will collect in the future

Introducing the Application Server Instance SMF Interval Statistics Workspace

Introduction

The Application Server Instance SMF Interval Statistics workspace provides SMF statistical information about activity experienced by an application server instance. You can access this type of statistical information for both J2EE and MOFW servers. These workspaces provide communication and traffic information about the application server instance.

Accessing the Application Server Instance SMF Interval Statistics workspace

Use these steps to access the Application Server Instance SMF Interval Statistics workspace.

1. Expand the Navigator.
2. Expand the WebSphere Application Server OS/390 node to access its nested items.
3. Select J2EE Server or MOFW Server you want to inspect and expand its Navigator tree.
4. Select the application server instance you want to inspect and expand its Navigator tree.
5. Choose the Application Server Instance SMF Interval Statistics selection in the Navigator.

Inside the Application Server Instance SMF Interval Statistics workspace

The predefined Application Server Instance SMF Interval Statistics workspace for a J2EE or MOFW server contains

- a plot chart that shows the number of current local and remote communication sessions during the interval
- a plot chart that shows the number of bytes local and remote communication sessions sent and received during the interval
- an Application Server Instance SMF Interval Statistics table view that reports SMF statistical data on server activity

This workspace also provides a direct link to the Application Server Instance SMF Interval Statistics History workspace.

Note: *Move the cursor over different areas of a chart to display flyover details about the items in the chart.*

Use

The Application Server Instance SMF Interval Statistics workspace provides SMF statistical information about communication sessions and byte traffic for the application server instance. Use this workspace for quick access to such information as the number of local and global transactions, the current and total number of communication sessions, and totals and rates for bytes sent and received from local and remote clients.

SMF information

This Application Server Instance SMF Interval Statistics workspace reports information collected from SMF record 120, subtype 3—the WebSphere Server Interval record.

Information presented in the Application Server Instance SMF Interval Statistics table view

The Application Server Instance SMF Interval Statistics table view displays data provided by the Application Server Instance SMF Interval Statistics attributes. The following is a list of the Application Server Instance SMF Interval Statistics attributes that report the data for this workspace.

- Bytes Received from Clients
- Bytes Received from Clients Rate
- Bytes Received from Local Clients
- Bytes Received from Local Clients Rate
- Bytes Received from Remote Clients
- Bytes Received from Remote Clients Rate
- Bytes Sent to Clients
- Bytes Sent to Clients Rate
- Bytes Sent to Local Clients
- Bytes Sent to Local Clients Rate
- Bytes Sent to Remote Clients
- Bytes Sent to Remote Clients Rate
- Current Communication Sessions
- Current Local Communication Sessions
- Current Remote Communication Sessions
- Global Transactions
- Interval Time
- Local Transactions
- Origin Node
- Host Name
- Sample Date and Time
- Server Instance Name
- Server Name
- Total Communication Sessions
- Total Local Communication Sessions
- Total Remote Communication Sessions

For detailed information about these attributes and their values, refer to the online Help for OMEGAMON XE for WebSphere Application Server for OS/390.

Introducing the J2EE Server Containers Workspace

Introduction

The J2EE Server Containers workspace reports detailed SMF information about any containers for an application server in which there was any activity during the interval.

Accessing the J2EE Server Containers workspace

Use these steps to access the J2EE Server Containers workspace.

1. Expand the Navigator.
2. Expand the WebSphere Application Server OS/390 node to access its nested items.
3. Select the J2EE Server item and expand its Navigator tree.
4. Select the application server instance you want to inspect and expand its Navigator tree.
5. Choose the Application Server Instance SMF Interval Statistics selection in the Navigator.

Inside the J2EE Server Containers workspace

The predefined J2EE Server Containers workspace contains

- a Methods Invocations bar chart that shows method invocations and EJB lifecycle events, such as, load, store, activation, and passivation during the interval
- a J2EE Server Containers table view that displays information about each container that experienced activity during the interval

This workspace also provides direct links to the J2EE Server Containers and J2EE Server Beans workspaces.

Note: *Move the cursor over different areas of the chart to display flyover details about the items in the chart.*

Use

The J2EE Server Containers table view provides statistical information about the each container that experienced activity during the interval in a J2EE application server instance.

Use this workspace for quick access to aggregate information about entity beans and bean methods for entity beans, stateless beans, and stateful beans within a container. For example, you can view totals and rates for bean method invocations, activations, passivations, stores, and loads.

SMF information

This J2EE Server Containers workspace reports information collected from SMF record 120, subtype 6—the WebSphere J2EE Container Interval record.

Information presented in the J2EE Server Containers table view

The J2EE Server Containers table view displays data provided by the J2EE Server Containers attributes. The following is a list of the J2EE Server Containers attributes that report the data for this workspace

- Activation Rate
- Activations
- Bean Methods
- Beans
- Container
- Entity Bean Activation Rate
- Entity Bean Activations
- Entity Bean Method Invocation Rate
- Entity Bean Method Invocations
- Entity Bean Passivation Rate
- Entity Bean Passivations
- Host Name
- Interval Time
- Load Rate
- Loads
- Method Invocation Rate
- Method Invocations
- Origin Node
- Passivation Rate
- Passivations
- Sample Date and Time
- Server Instance Name
- Server Name
- Stateful Activation Rate
- Stateful Activations
- Stateful Method Invocation Rate
- Stateful Method Invocations
- Stateful Passivation Rate
- Stateful Passivations
- Stateless Activation Rate
- Stateless Activations
- Stateless Method Invocation Rate
- Stateless Method Invocations
- Stateless Method Passivation Rate
- Stateless Method Passivations
- Store Rate
- Stores

For detailed information about these attributes and their values, refer to the online Help for OMEGAMON XE for WebSphere Application Server for OS/390.

Introducing the J2EE Server Beans Workspace

Introduction

The J2EE Server Beans workspace reports specific System Management Facility information about Enterprise Java beans.

Accessing the J2EE Server Beans workspace

Use these steps to access the J2EE Server Beans workspace.

1. Expand the Navigator.
2. Expand the WebSphere Application Server OS/390 node to access its nested items.
3. Select the J2EE Server item and expand its Navigator tree.
4. Select the application server instance you want to inspect and expand its Navigator tree.
5. Choose the Application Server Instance SMF Interval Statistics selection in the Navigator.

Inside the J2EE Server Beans workspace

The predefined J2EE Server Beans workspace contains

- a bar chart that shows the number of method invocations during the interval
- a J2EE Server Beans table view that displays statistical information about each active Enterprise Java bean during the interval

This workspace also provides a direct link to the J2EE Server Bean Methods workspace.

Note: *Move the cursor over different areas of the chart to display flyover details about the items in the chart.*

Use

The J2EE Server Beans table view provides statistical data about each Enterprise Java Bean (EJB) deployed to a J2EE application server instance. This information includes some configuration data, such as bean type and reentrance policy, and aggregate information about methods within the EJB, such as, totals and rates for method invocations and bean activations, passivations, and stores.

Use this workspace for quick access to configuration data about deployed Java beans and aggregate information about methods within each EJB, such as, totals and rate for method invocations and EJB activations, passivations, stores, and loads. For example, consider removing explicit stateful session beans if they are causing excessing activations and passivations that impair performance.

SMF information

This J2EE Server Containers workspace reports information collected from SMF record 120, subtype 6—the WebSphere J2EE Container Interval record.

Information presented in the J2EE Server Beans table view

The environmental variables table views display data provided by the J2EE Server Beans attributes. The following is a list of these attributes.

- Activation Rate
- Activations
- Bean Methods
- Bean Name
- Bean Type
- Container
- Host Name
- Interval Time
- Load Rate
- Loads
- Method Invocation Rate
- Method Invocations
- Origin Node
- Passivation Rate
- Passivations
- Reentrance Policy
- Sample Date and Time
- Server Instance Name
- Server Name
- Store Rate
- Stores

For detailed information about these attributes and their values, refer to the online Help for OMEGAMON XE for WebSphere Application Server for OS/390.

Introducing the J2EE Server Bean Methods Workspace

Introduction

The J2EE Server Bean Methods workspace reports System Management Facility information about EJB methods.

Accessing the J2EE Server Bean Methods workspaces

Use these steps to access the J2EE Server Bean Methods workspaces.

1. Expand the Navigator.
2. Expand the WebSphere Application Server OS/390 node to access its nested items.
3. Select the J2EE Server item and expand its Navigator tree.
4. Select the application server instance you want to inspect and expand its Navigator tree.
5. Choose the Application Server Instance SMF Interval Statistics selection in the Navigator.

Inside the J2EE Server Bean Methods workspace

The predefined J2EE Server Bean Methods workspace for a J2EE server contains

- a bar chart that shows the 10 worst average response times for bean methods during the interval
- a bar chart that shows the 10 highest invocation rates for bean methods during the interval
- a J2EE Server Bean Methods table view that reports SMF statistical data about each bean method invoked during the interval

Note: *Move the cursor over different areas of a chart to display flyover details about the items in the chart.*

Use

The J2EE Server Bean Methods table view reports statistical information about each Java bean method invoked within the J2EE application server instance.

Use this workspace for quick access to such information as transaction policy, ejbRoles, and non-framework exceptions. You can also view the total, rate, average execution times, and maximum execution times for method invocations, activations, passivations, and stores.

SMF information

This J2EE Server Containers workspace reports information collected from SMF record 120, subtype 6—the WebSphere J2EE Container Interval record.

Information presented in the J2EE Server Bean Methods table view

The J2EE Server Bean Methods table view displays data provided by the J2EE Server Bean Methods attributes. A list of these attributes appears on the next page.

The J2EE Server Bean Methods attributes that report the data for this workspace.

- Activation Rate
- Activations
- Average Activate Execution Time
- Average Load Execution Time
- Average Passivate Execution Time
- Average Response Time
- Average Store Execution Time
- Bean Name
- Container
- EJB Roles
- Host Name
- Interval Time
- Invocation Rate
- Invocations
- Load Rate
- Loads
- Maximum Activate Execution Time
- Maximum Load Execution Time
- Maximum Passivate Execution Time
- Maximum Response Time
- Maximum Store Execution Time
- Method Arguments
- Method Name
- Origin Node
- Passivation Rate
- Passivations
- Sample Date and Time
- Server Instance Name
- Server Name
- Store Rate
- Stores
- Transaction Policy

For detailed information about these attributes and their values, refer to the online Help for OMEGAMON XE for WebSphere Application Server for OS/390.

Introducing the MOFW Server Containers Workspace

Introduction

The MOFW Server Containers workspace reports SMF information about active containers for a MOFW server.

Accessing the MOFW Server Containers workspace

Use these steps to access the MOFW Server Containers workspace.

1. Expand the Navigator.
2. Expand the WebSphere Application Server OS/390 node to access its nested items.
3. Select the MOFW Server item and expand its Navigator tree.
4. Select the MOFW application server instance you want to inspect and expand its Navigator tree.
5. Choose the MOFW Server Containers selection in the Navigator.

Inside the MOFW Server Containers workspace

The predefined MOFW Server Containers workspace contains

- a bar chart that shows how many methods were invoked in the container during the interval
- a bar chart that shows the number of passivations per second for class instances during the interval
- a MOFW Server Containers table view that reports SMF-statistics for each each container that experienced activity during the interval

This workspace also provides a direct link to the MOFW Server Containers History workspace.

Note: *Move the cursor over different areas of the chart to display flyover details about the items in the chart.*

Use

The MOFW Server Containers table view provides statistical information about each container that experienced any activity in a MOFW application server instance during an interval.

Use this workspace for quick access to such information as transaction and security policies. You can also view the aggregate information about all classes within the container, such as, totals and rates for creates, activations, removes, and passivations for a class instance. You can also view totals and rates for method invocations.

SMF information

This MOFW Server Containers workspace reports information collected from SMF record 120, subtype 4—the WebSphere Container Interval record.

Information presented in the MOFW Server Containers table view

The MOFW Server Containers table view displays data provided by the MOFW Server Containers attributes. The following is a list of the MOFW Server Containers attributes that report the data for this workspace

- Class Instance Activate Rate
- Class Instance Activations
- Class Instance Create Rate
- Class Instance Creates
- Class Instance Passivate Rate
- Class Instance Passivations
- Class Instance Remove Rate
- Class Instance Removes
- Classes
- Container
- Host Name
- Interval Time
- Method Invocation Rate
- Method Invocations
- Methods
- Origin Node
- Sample Date and Time
- Server Instance Name
- Server Name
- Transaction Policy

For detailed information about these attributes and their values, refer to the online Help for OMEGAMON XE for WebSphere Application Server for OS/390.

Introducing the MOFW Server Classes Workspace

Introduction

The MOFW Server Classes workspace reports SMF information about classes associated with a MOFW server.

Accessing the MOFW Server Classes workspace

Use these steps to access the MOFW Server Classes workspace.

1. Expand the Navigator.
2. Expand the WebSphere Application Server OS/390 node to access its nested items.
3. Select the MOFW Server item and expand its Navigator tree.
4. Select the MOFW application server instance you want to inspect and expand its Navigator tree.
5. Choose the MOFW Server Containers selection in the Navigator.

Inside the MOFW Server Classes workspace

The predefined MOFW Server Classes workspace contains

- a bar chart that shows how many methods were invoked in the container during the interval
- a bar chart that shows the rate of class instance passivations based upon the number of events during the interval
- a MOFW Server Classes table view that displays statistical information about each class associated with the MOFW server during the interval

Note: *Move the cursor over different areas of the chart to display flyover details about the items in the chart.*

Use

The MOFW Server Classes table view provides SMF statistical data about classes associated with a MOFW server.

Use this workspace for quick access to totals and rates for creates, activations, removes, and passivations. You can also view the method activation total and rate.

SMF information

The MOFW Server Classes workspace reports information collected from SMF record 120, subtype 4—the WebSphere Container Interval record.

Information presented in the MOFW Server Classes table view

The environmental variables table views display data provided by the MOFW Server Classes attributes. The following is a list of these attributes.

- Class Name
- Container
- Host Name
- Instance Activate Rate
- Instance Create Rate
- Instance Passivate Rate
- Instance Remove Rate
- Instances Activated
- Instances Created
- Instances Passivated
- Instances Removed
- Interval Time
- Method Invocation Rate
- Method Invocations
- Methods
- Origin Node
- Sample Date and Time
- Server Instance Name
- Server Name

For detailed information about these attributes and their values, refer to the online Help for OMEGAMON XE for WebSphere Application Server for OS/390.

Introducing the MOFW Server Methods Workspace

Introduction

The MOFW Server Methods workspace provides information about each method in a class that was invoked for a MOFW server. SMF recording must be enabled for the OMEGAMON XE agent to report data in this workspaces.

Accessing the MOFW Server Methods workspaces

Use these steps to access the MOFW Server Methods workspaces.

1. Expand the Navigator.
2. Expand the WebSphere Application Server OS/390 node to access its nested items.
3. Select the MOFW Server item and expand its Navigator tree.
4. Select the MOFW application server instance you want to inspect and expand its Navigator tree.
5. Choose the MOFW Server Containers selection in the Navigator.

Inside the MOFW Server Methods workspace

The predefined MOFW Server Methods workspace for a MOFW server contains

- a bar chart that shows the 10 worst response times for bean methods during the interval
- a bar chart that shows the 10 highest invocations per second for bean methods during the interval
- an MOFW Server Methods table view that reports SMF statistical data about each method that was invoked during for a class during the interval

Note: *Move the cursor over different areas of a chart to display flyover details about the items in the chart.*

Use

The MOFW Server Methods table view reports SMF statistical information about each method of a class invoked within a MOFW server with activity.

Use this workspace for quick access to such information as invocation totals and rates, non-framework exceptions, and average and maximum response times.

SMF information

This MOFW Server Containers workspace reports information collected from SMF record 120, subtype 4—the WebSphere Container Interval record.

Information presented in the MOFW Server Methods table view

The MOFW Server Methods table view displays data provided by the MOFW Server Methods attributes. The following is a list of these attributes.

- Average Response Time
- Class Name
- Container
- Host Name
- Interval Time
- Invocation Rate
- Invocations
- Maximum Response Time
- Method Name
- Non-framework Exceptions
- Origin Node
- Sample Date and Time
- Server Instance Name
- Server Name

For detailed information about these attributes and their values, refer to the online Help for OMEGAMON XE for WebSphere Application Server for OS/390.

8

JVM Threads, Methods, and Monitor Contention

Overview

This chapter provides instructions for accessing the JVM Threads workspace and its related workspaces for JVM Contention, JVM Methods, and the JVM Method Summary workspace. It also discusses the contents of these workspaces that provide Java Virtual Machine (JVM) data.

Chapter contents

Background Information

Chapter goals

This chapter introduces you to the information that is available in OMEGAMON XE for WebSphere Application Server to track performance associated with JVM threads and methods. It relates this information to the configuration parameters that you can use to manage the collection of the data through the Candle profiler.

Chapter navigator

The chart below shows where you can locate information in this chapter about specific tasks you want to perform.

If you want to...	See...
identify workspaces that display detailed JVM data	page 145
find out how to collect JVM data in WebSphere Application Server for OS/390	page 146
identify which configuration parameters are used to collect JVM data	page 146
find out how to start JVM data collection by the profiler	page 147
review the information presented in the JVM Threads workspace	page 148
review the information presented in the JVM Monitor Contention workspace	page 150
review the information presented in the JVM Methods workspace	page 152
review the information presented in the JVM Method Summary workspace	page 154

About JVM Threads, Methods, and Contention Information

Benefits

You can use information about JVM threads, methods, and monitor contention to monitor performance, spot problems, and evaluate the JVM implementation by an application. It can help you identify resource usage, delays, and heap allocations.

OMEGAMON XE for WebSphere Application Server for OS/390 lets you select the specific data that will be collected and when. This avoids the drawbacks associated with instrumentation within an application or using pre-defined situations that may not cover production issues that should be monitored.

Monitoring thread information with the OMEGAMON XE agent

OMEGAMON XE for WebSphere Application Server for OS/390 uses non-intrusive data collection by letting you define configuration parameters to collect the specific type of data you want and then use the JVM profiler to turn data collection on and off as you choose. To avoid unwanted overhead, the JVM profiler and data collection are turned off by default.

The workspaces that give you a direct view of this thread-related data are the

- JVM Threads
- JVM Methods
- JVM Monitor Contention
- JVM Method Summary

These workspaces and the attributes that support them are described later in this chapter.

Note: *The JVM profiler must be active for the OMEGAMON XE agent to report data in these workspaces.*

Collecting the JVM Thread Information You Want

Overview

To collect the type of JVM thread data you want, you need to manage the OMEGAMON XE WebSphere agent parameters and the JVM profiler that control its reporting. This topic supplies general information about how you set up this data collection process.

Configuration parameters for the OMEGAMON XE agent

Use configuration settings in the KWWXML member to manage collection of the data by the JVM profiler. The settings that you use for the agent parameters and application server parameters in this member will tailor the collection of data to meet your needs. Rather that just delivering a quantity of general data, these parameters enable you to gather the specific data in which you are interested.

The following table summarizes the functions of these configuration parameters. For detailed information about using each of these parameters and the KWWXML member, see Chapter 2, "Configuring OMEGAMON XE for WebSphere Application Server for OS/390" on page 39.

Table 4. Summary of JVM Configuration Parameters

Parameter	Function
OMEGAMON XE for WebSphere Agent Parameters	
DefaultJVMProfilerCollect	Specifies the type of statistical data to be collected by the JVM profiler
DefaultJVMProfilerCollectClasses	Identifies the Java classes for which method, contention, and heap data should be collected
DefaultJVMProfilerCollectStopAfter	Determines how often, in milliseconds, that CPU sampling should be performed within the JVM

Table 4. Summary of JVM Configuration Parameters (continued)

Parameter	Function
DefaultJVMProfilerIgnoreMethods	Identifies the names of Java methods for which CPU busy samples should be ignored by the profiler
DefaultJVMSampleInterval	Determines how often, in seconds, JVM profiler data is transmitted back to the KWWAGENT address space
Application Server Parameters	
JVMProfilerCollect	Specifies the type of statistical data to be collected by the JVM profiler
JVMProfilerCollectClasses	Identifies the Java classes for which method, contention, and heap data should be collected
JVMProfilerCollectStopAfter	Determines how often, in milliseconds, that CPU sampling should be performed within the JVM
JVMProfilerIgnoreMethods	Identifies the names of Java methods for which CPU busy samples should be ignored by the profiler
JVMSampleInterval	Determines how often, in seconds, JVM profiler data is transmitted back to the KWWAGENT address space

Activating the profiler

After you define the configuration settings for collecting JVM data in the KWWXML member, you will need to activate the profiler and recycle the application server when you want to collect the JVM data. To do this, use the WAS390 JVMSTAT - J2EE take action command. See “Activating the JVM Profiler” on page 66 for these instructions.

Introducing JVM Threads Information

Overview

The JVM Threads workspace reports data from the profiler about monitor contention within the JVMs.

Accessing the JVM Threads workspace

Use these steps to access the JVM Threads workspace.

1. Expand the Navigator.
2. Expand the WebSphere Application Server OS/390 node to access its nested items.
3. Select the J2EE Server item and expand its Navigator tree.
4. Select the application server you want to inspect and expand its Navigator tree.
5. Choose the JVM Threads selection in the Navigator.

Inside the JVM Threads workspace

The predefined JVM Threads workspace contains

- a bar chart that shows which ten threads used the largest percentage of CPU resources during the sampling interval
- a bar chart that shows which ten threads waited the longest to enter the monitor or use an object during the sampling interval
- a JVM Threads table view that displays information associated with wait times, heap activity, CPU usage, and the profiler

Note: *Move the cursor over different areas of the bar charts to display flyover details about the items in a chart.*

From this workspace, you can also access the JVM Methods and JVM Monitor Contention workspaces.

JVM Threads Table View

Use

This workspace provides information from the profiler about the threads running in a Java Virtual Machine that is hosting an application server. The JVM Threads workspace reports information about activity for JVM Threads.

Certain data collection is turned off by default because of high overhead. If you want to report this data in the workspace, then activate the JVM profiler. See "Activating the JVM Profiler" on page 66 for this information.

In the table view, you can also select a row to link to the JVM Methods and JVM Monitor Contention workspaces.

Information presented in the JVM Threads table view

The following is a list of JVM Threads for OS/390 attributes reported in the JVM Threads table view.

- Bytes Allocated
- CPU Busy Samples
- CPU Busy Samples Taken
- CPU Milliseconds Used by Application Server
- Heap Allocation Rate
- Heap Allocations
- Interval Time
- Monitor Contention CPU Samples
- Monitor Contention Delay Time
- Monitor Contention Delays
- Monitor Wait CPU Samples
- Node Name
- Object Wait Delay Time
- Object Wait Delays
- Origin Node
- Parent Name
- Percent CPU Used
- Percent CPU Used by Application Server
- Percent CPU Used by Thread
- Percent Waiting for Object
- Percent Waiting to Enter Monitor
- Process ID
- Sample Date and Time
- Server Instance Name
- Thread Group Name
- Thread ID
- Thread Name
- Total Sampler Cycles

For detailed information about these attributes and their values, refer to the online Help for OMEGAMON XE for WebSphere Application Server for OS/390.

Introducing JVM Monitor Contention Information

Overview

The JVM Monitor Contention workspace reports data from the profiler about monitor contention within the JVMs.

Accessing the JVM Monitor Contention workspace

Use these steps to access the JVM Monitor Contention workspace.

1. Expand the Navigator.
2. Expand the WebSphere Application Server OS/390 node to access its nested items.
3. Select the J2EE Server item and expand its Navigator tree.
4. Select the application server you want and expand its Navigator tree.
5. Choose the JVM Threads selection in the Navigator.
6. Click the link beside a row in the JVM Threads table and choose the JVM Contention workspace from the pop-up menu.

Inside the JVM Monitor Contention workspace

The predefined JVM Monitor Contention workspace contains

- a bar chart that shows which the percentage of contention delay for monitor object classes
- a JVM Monitor Contention table view that displays information about contention, delays, times and the names of the monitor object class, waiting class, waiting method, and waiting method signature

Note: *Move the cursor over different areas of the bar chart to display flyover details about the items in the chart.*

From this workspace, you can also access the JVM Threads and JVM Methods workspaces.

JVM Monitor Contention Table View

Use

This workspace provides information from the profiler about the threads running in a Java Virtual Machine that is hosting an application server. The JVM Monitor Contention workspace reports information about activity for JVM Monitor Contention.

Certain data collection is turned off by default because of high overhead. If you want to report this data in the workspace, then activate the Candle profiler. See "Activating the JVM Profiler" on page 66 for this information.

Information presented in the JVM Monitor Contention table view

The JVM Monitor Contention for OS/390 attributes report the data in this workspace. The following is a list of the attributes reported in the JVM Monitor Contention table view.

- Contention Delay
- Contention Events
- Interval Time
- JVM Monitor Contention
- Monitor Object Class Name
- Node Name
- Object Wait Delay Time
- Object Wait Delays
- Origin Node
- Percent Contention Delay
- Percent Waiting for Object
- Process ID
- Sample Date and Time
- Server Instance Name
- Thread ID
- Waiting Class Name
- Waiting Method Name
- Waiting Method Signature

For detailed information about these attributes and their values, refer to the online Help for OMEGAMON XE for WebSphere Application Server for OS/390.

Introducing JVM Methods Information

Overview

The JVM Methods workspace reports data from the profiler about monitor contention within the JVMs.

Accessing the JVM Methods workspace

Use these steps to access the JVM Methods workspace.

1. Expand the Navigator.
2. Expand the WebSphere Application Server OS/390 to access its nested items.
3. Select the J2EE Server item and expand its Navigator tree.
4. Select the application server you want and expand its Navigator tree.
5. Choose the JVM Threads selection in the Navigator.
6. Right-click a row in the JVM Threads table and choose the JVM Methods workspace from the pop-up menu.

Inside the JVM Methods workspace

The predefined JVM Methods workspace contains

- a bar chart that shows which ten threads used the largest percentage of CPU resources during the sampling interval
- a bar chart that shows which ten threads waited the longest to enter the monitor or use an object during the sampling interval
- a JVM Methods table view that displays information about wait times associated with a monitor or object, CPU usage, the number of samples taken by the CPU, and heap activity

Note: *Move the cursor over different areas of the bar charts to display flyover details about the items in a chart.*

From this workspace, you can also access the JVM Threads and JVM Monitor Contention workspaces.

JVM Methods Table View

Use

This workspace provides information from the profiler about the methods being used in the Java Virtual Machine that is hosting an application server. It reports activity and performance data.

Certain data collection is turned off by default because of high overhead. If you want to report this data in the workspace, then activate the JVM profiler. See "Activating the JVM Profiler" on page 66 for this information.

Information presented in the JVM Methods table view

The JVM Methods table view displays data provided by the JVM Methods for WAS OS/390 attributes. The following is a list of attributes reported in the JVM Methods table.

- Bytes Allocated
- Class Name
- Contention Delays
- CPU Busy Samples Taken
- CPU Samples Directly Busy
- CPU Samples Indirectly Busy
- Heap Allocation Rate
- Heap Allocations
- Interval Time
- Method Name
- Method Signature
- Monitor Contention Delay Time
- Node Name
- Object Wait Delay Time
- Origin Node
- Percent CPU Indirectly Used
- Percent CPU Used
- Percent Waiting for Object
- Percent Waiting to Enter Monitor
- Process ID
- Sample Date and Time
- Server Instance Name
- Thread ID
- Total Sampler Cycles
- Wait Delays

For detailed information about these attributes and their values, refer to the online Help for OMEGAMON XE for WebSphere Application Server for OS/390.

Introducing JVM Method Summary Information

Overview

The JVM Method Summary workspace provides fast access from the Navigator to the performance data for the classes and methods being used in the Java Virtual Machine that is hosting an application server.

Accessing the JVM Method Summary workspace

Use these steps to access the JVM Method Summary workspace.

1. Expand the Navigator.
2. Expand the WebSphere Application Server OS/390 to access its nested items.
3. Select the J2EE Server item and expand its Navigator tree.
4. Select the application server you want and expand its Navigator tree.
5. Choose the JVM Method Summary selection in the Navigator.

Inside the JVM Method Summary workspace

The predefined JVM Method Summary workspace contains

- a bar chart that shows which 10 methods used the most CPU during the sampling interval
- a bar chart that shows which 10 methods waited the longest to enter a monitor or use an object during the sampling interval
- a JVM Method Summary table view that displays information about delays associated with each method, CPU usage, and heap activity

Note: *Move the cursor over different areas of the bar charts to display flyover details about the items in a chart.*

JVM Method Summary Table View

Use

This workspace provides information from the profiler about the methods being used in the Java Virtual Machine that is hosting an application server. It reports activity and performance data. The table view contains a row of data for each unique class and method name for which profiler data was collected. Each row contains a summary of all the activity for that class and method, combining data from various Java threads and recognizable class and method names.

Certain data collection is turned off by default because of high overhead. If you want to report this data in the workspace, then use the WAS390 JVMSTAT take action command to activate the JVM profiler. See "Activating the JVM Profiler" on page 66 for this information.

The JVM Method Summary workspace differs from the JVM Methods Information workspace in that the method usage data is summarized across all threads running within the JVM. The JVM Methods Information workspace can only be viewed after selecting a particular JVM thread, and only shows the method activity associated with the selected thread. The JVM Method Summary workspace can be selected directly from the Navigator tree, and shows all method activity, regardless of which thread the methods ran on.

Information presented in the JVM Method Summary table view

The JVM Method Summary table view displays data provided by the JVM Method Summary for WAS OS/390 attributes. The following is a list of attributes reported in the JVM Method Summary table.

- Bytes Allocated
- Class Name
- Contention delays
- CPU Busy sAmples Taken
- CPU Samples Directly Busy
- CPU Samples Indirectly Busy
- Heap Allocation Rate
- Heap allocaTions
- Interval Time
- Method Name
- Method Signature
- Monitor Contention Delay Time
- Node Name
- Object Wait Delay Time
- Origin Node
- Percent CPU Indirectly Used
- Percent CPU Used
- Percent Waiting For Object
- Percent Waiting To Enter Monitor
- Process ID
- Sample Date and Time
- Server Name
- Wait Delays

For detailed information about these attributes and their values, refer to the online Help for OMEGAMON XE for WebSphere Application Server for OS/390.

9

Workload Information for Performance Tuning

Overview

This chapter provides information about collecting workload performance data from J2EE application servers. It contains background information about using workloads and instructions for accessing the various workspaces that let you monitor this type of data.

Chapter contents

Background Information

Chapter goals

This chapter focuses on performance metrics that you can obtain by defining workloads for J2EE application servers. It covers the basic concepts associated with workloads, delays, and instrumentation that you use in collecting this specialized performance data.

It identifies the keywords that you use to set up workload instrumentation and take action commands that you use to turn on the collection of this data. This chapter also describes the workspaces in which you can view workload data and provides instructions for accessing them

Chapter navigator

The chart below shows where you can locate information in this chapter about specific tasks you want to perform.

If you want to...	**See...**
become familiar with using workloads and concepts you need to know,	page 159
review the steps for collecting workload information,	page 162
identify take action commands that you use to turn data collection on and off,	page 162
learn about keywords to use in the instrumentation control file,	page 163
identify workspaces that display workload data,	page 166
learn about data available for all workloads,	page 167
learn about data available for selected workloads,	page 170
learn about data available for selected delays,	page 173
learn about data available for the 10 most frequently used workloads	page 176
learn about data available for transactions that exceed response time limits,	page 178
learn about data available for database response times,	page 182
learn about data available for HTTP sessions for specific browsers,	page 184
learn about data available for messaging queues and MQSeries,	page 186
learn how you can trace instrumented workloads and the data you can obtain,	page 188

About Workload Information

Benefits

You can use workload information to monitor performance, spot problems, and evaluate the JVM implementation by an application. It can help you identify issues associated with resource usage, delays, and heap allocations.

For example, in OMEGAMON XE for WebSphere Application Server for OS/390, you may see that a servlet is taking twice as long to complete its operation as usual. By using workload analysis, you can identify the cause of major delays. You may find that the servlet is using excessive time to perform an SQL request against a database, locate an entity bean, or invoke an entity bean method. With this information, you can target your tuning to the areas that cause the most delays and degradation.

The instrumentation provided by OMEGAMON XE for WebSphere Application Server for OS/390 lets you avoid the drawbacks associated with instrumentation within an application or using pre-defined situations that may not cover production issues that should be monitored.

Collecting information

You can use workload analysis to determine where an application is spending it's time. This data is collected and presented in terms of J2EE functional areas (for example JDBC, JNDI, or EJB Methods). This information is collected by the Candle Workload Analysis component through a technique called dynamic object code instrumentation.

As Java classes are loaded into a WebSphere application server region, the workload analysis dynamically inserts data collection hooks directly into the object code. As the Java object code is subsequently executed, those data collection hooks will pick up the requested workload analysis data and transport the data back to KWWAGENT, where it can be displayed, stored, and used for situation monitoring.

Monitoring workloads with the OMEGAMON XE agent

OMEGAMON XE for WebSphere Application Server for OS/390 uses non-intrusive data collection by letting you set up instrumentation in the

OMEGAMON XE agent that will let you collect the specific type of data you want and also turn data collection on and off as you choose. To avoid unwanted overhead, the collection of any workload data is turned off by default.

Although dynamic object code instrumentation can only be accomplished as Java classes are being loaded at startup time, the data collection hooks that are inserted are initially inactive (meaning that they collect no data, and therefore, use little or no overhead). Only when the hooks activated by KWWAGENT, either through agent startup parameters, or with a take action command, will the data collection hooks become active and collect workload data. In this way, the overhead associated with workload data collection is controlled easily and dynamically, with no requirement to restart application servers.

Key concepts and terminology

These are the basic concepts and terms you need to know to use workload analysis.

Workload

This is a unit of work whose response time directly reflects the response time perceived by an end user. For the OMEGAMON XE agent, a workload is either a servlet or EJB method. You can define and monitor workloads as needed to track their performance on your system. Similar to a transaction, the execution of a workload has a response time associated with it. This is the amount of real time that it took for the workload to complete and a means of evaluating performance.

For the OMEGAMON XE agent, a workload is identified by the combination of the workload type and its workload name. For example, a servlet workload type plus its servlet name or EJB class and method name would identify a particular workload.

Delay

This is any measurable factor that contributes to the response time of a workload. For example, the time it takes to read a row of data from an SQL database is a delay. A delay can also be another workload. For example, if a servlet locates and calls an EJB method, the EJB method becomes a workload that contributes to the response time of the servlet.

Instrumentation

This is the practice of modifying program code so that the amount of time it takes the code to run, along with other statistics, can be measured. By setting up instrumentation in the OMEGAMON XE agent instead of your applications, you can collect this data more quickly and efficiently,

Collecting the Workload Information You Want

Overview

This topic provides an overview of the instrumentation process and familiarizes you with basic concepts you will need to know to collect workload information.

Basic steps for instrumenting workloads for the agent

To collect the type of workload data you want, you need to perform these basic tasks

1. Customize the kwwiuser.xml control file to define the classes you want to instrument.

   You will need to change the values for the parameters and keywords in this instrumentation file whenever you want to track new workloads or change the type data you are collecting.

   Note: *If you use several*

2. Run the setup CandleWAEnabler.sh shell script that set up the Candle workload analysis Class Loader Jar file. Generally, you will only need to rerun this shell script if maintenance is applied either to the OMEGAMON XE for WebSphere product, or to the IBM-provided Java Development Kit (JDK).

3. Use the IBM System Management User Interface (SMEUI) to change the configuration of the IBM WebSphere Application Server so that the Candle Workload Analysis component will be automatically loaded and run.

4. Use the take action commands to control the collection of workload data.

   A. Use the WAS390 Dynamic Workload Analysis take action command to turn the collection of workload data on and off.

   B. Use the WAS390 Set Workload Exception Thresholds take action command to define response time limits that will report long-running transactions.

   For details on instrumenting workloads, see "Setting Up the Workload Analysis Control File" on page 72 in Chapter 2, Configuring OMEGAMON XE for WebSphere Application Server for OS/390.

Keywords for your instrumentation control file

Use keywords in the kwwiuser.xml file to implement collection of workload data. The settings that you use in the instrumentation file will tailor the collection of data to meet your needs. Rather that just delivering a quantity of general data, these keywords enable you to gather the specific data in which you are interested.

The following table summarizes the functions of these instrumentation keywords. For detailed information about using each of these keywords and preparing the OMEGAMON XE agent for workload analysis, see "Setting Up the Workload Analysis Control File" on page 72 in Chapter 2, Workload Information for Performance Tuning.

Table 5. Summary of Instrumentation Keywords in Kwwiuser.xml

Keyword	Function
For KWWINSTR entries…	
Version	Identifies the version of the OMEGAMON XE product
MaxClasses	Identifies the maximum number of classes to instrument
CandleRoot	Identifies the name of the Candle root directory
LogFileName	Indicates the name of the log file if detailed tracing is enabled
SysInstr	Indicates whether or not instrumentation will be performed directly or indirectly
JVMProfilingPort	Indicates the IP port number
TraceActive	Indicates whether or not detailed tracing should be enabled
CollectCPUTime	Indicates whether or not CPU usage data is to be collected
ShowInternalWorkloads	Indicates how workloads invoked by other workloads should be treated as a separate workload or a delay

Table 5. Summary of Instrumentation Keywords in Kwwiuser.xml

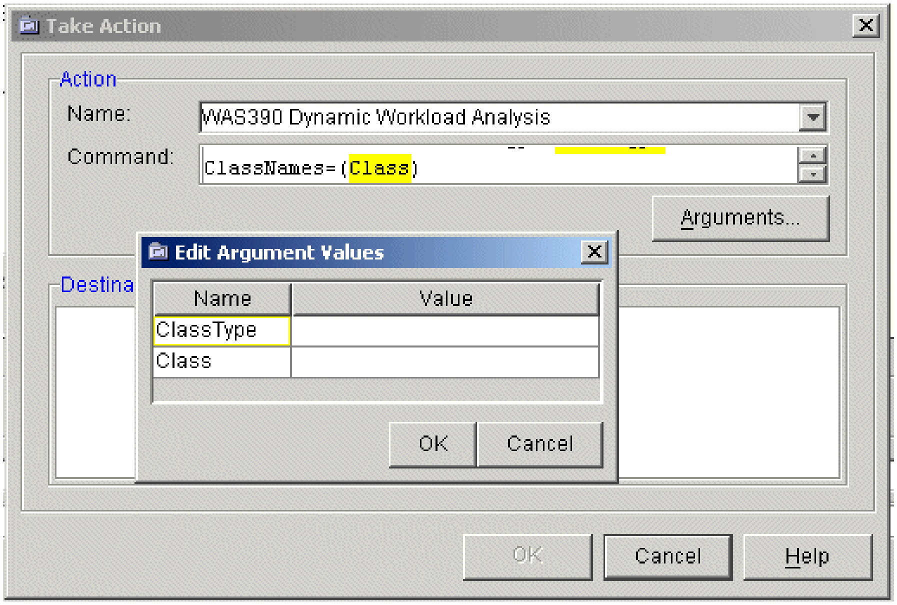
Keyword	Function
For Class entries…	
Name	Provides a mask to identify the names of the class or classes for the entry
ClassType	Indicates the type of class being identified by this statement
MethodType	Indicates the type of methods that should be instrumented
MethodNames	Provides a mask that identifies the names of the methods that are to be instrumented in addition to those identified by the MethodType parameter
SysIntr	Indicates whether or not instrumentation will be performed directly or indirectly
ActiveByDefault	Indicates whether or not data collection will be initially active
ScanForIndirect	Indicates whether or not methods in this class should be scanned for calls to methods that are being indirectly instrumented

Starting workload data collection

Perform these steps to start an start collecting workload data.

1. In the CandleNet Portal Navigator, select any J2EE workspace that contains workload information you want to trace, for example, Databases. Right-click and choose **Take Action** from the pop-up menu.

2. In the Take Action dialog, scroll the commands in the Name field and choose **WAS390 Dynamic Workload Analysis** to display the following dialog.



Name: WAS390 Dynamic Workload Analysis

Command: ClassNames=(Class)

Name	Value
ClassType	
Class	

3. Complete the Value fields described below.

Table 6. WAS390 Dynamic Workload Analysis Command Values

Parameter Name	Explanation
Class_Type	Provide one of the following values to specify type of class you want to trace: **user** or **system**.
Class	Provide the name of the instrumented class that you want to trace. You can provide multiple class masks, separated by commas or semi-colons. For example, a mask of **ClassName=*** specifies that trace data will be collected for all classes that match the workload type.

Workspaces that Display Workload Metrics

Overview

This topic introduces you to the workspaces in OMEGAMON XE for WebSphere Application Server for OS/390 that report workload information in graphic and table view form. In using the graphs, move the cursor over different areas to display flyover details about the items.

Note: *These workspaces will not contain data if you have not configured the agent to collect this data and started data collection with the WAS390 Dynamic Workload Analysis take action command.*

Viewing workload data in workspaces

The following is a list of workspaces that contain graphic and table views of collected workload data. The list is indented to identify workspaces that are accessed from primary workspaces that are listed in the Navigator.

- All Workloads
  - Ten Most Frequently Used Workloads
  - Selected Workload - Current
  - Selected Workload - History
  - Selected Workload Delays
  - Selected Workload Delay - History
  - Longest Running Workloads
  - Longest Running Workloads - History
- Selected Longest Running Workload
  - Selected Longest Running Workload - History
- Datasources
  - Selected Datasource - History
- HTTP Sessions
- JMS Summary
  - Selected Queue - History
- Application Trace Files
- Application Trace

These workspaces are described by the following topics in this chapter.

Introducing the All Workloads Information

Overview

The All Workloads workspace reports summary data pertaining to response times for instrumented workloads that were executed during the current interval.

Accessing the All Workloads workspace

Use these steps to access the All Workloads workspace.

1. Expand the Navigator.
2. Expand the WebSphere Application Server OS/390 node to access its nested items.
3. Select the J2EE Server item and expand its Navigator tree.
4. Select the application server you want and expand its Navigator tree.
5. Choose the All Workloads selection in the Navigator.

Inside the All Workloads workspace

The predefined All Workloads workspace contains

- a bar chart that shows the ten worst response times for workloads
- an All Workloads - Current Interval table view that displays information about a variety of delays and their response time measurements.

From this workspace, you can also access subordinate workspaces for Selected Workload, Selected Workload History, Selected Workload Delay, Selected Workload Delay History, and Ten Most Frequently Used Workloads.

Use

The All Workloads workspace provides performance information about response times for all workloads being instrumented on the application server. In the bar chart, you can quickly identify workloads with the worst response times and see what types of delays were consuming the most time for each of them. Important performance measurements in this workspace are the rate of occurrence, the average response time, and the percent of time the workload spent performing a specific activity.

Information presented in the All Workloads table view

The following page provides a list of the attributes reported in the All Workloads table view.

These are the Workload Degradation Summary for WAS OS/390 attributes used to report metrics for the All Workloads workspace.

- Average CPU Time
- Average Time
- Class Name
- EJB Home Delay Percent
- EJB Home Delay Time
- EJB Method Delay Percent
- EJB Method Delay Time
- EJB Remote Delay Percent
- EJB Remote Delay Time
- Interval Time
- JMS Delay Percent
- JMS Delay Time
- JNDI Delay Percent
- JNDI Delay Time
- JTA Delay Percent
- JTA Delay Time
- Max Time
- Method Name
- Miscellaneous Delay Percent
- Miscellaneous Delay Time
- Network Socket Delay Percent
- Network Socket Delay Time
- Node Name
- Number EJB Home Delays
- Number EJB Method Delays
- Number EJB Remote Delays
- Number JMS Delays
- Number JNDI Delays
- Number JTA Delays
- Number Miscellaneous Delays
- Number Network Socket Delays
- Number of Occurrences
- Number Servlet Delays
- Number SQL Connection Delays
- Number SQL Query Delays
- Number SQL Update Delays
- Number User-defined Delays
- Origin Node
- Process ID
- Rate of Occurrence
- Sample Date and Time
- Server Instance Name
- Server Name
- Servlet Delay Percent
- Servlet Delay Time
- SQL Connection Delay Percent
- SQL Connection Delay Time
- SQL Query Delay Percent
- SQL Query Delay Time
- SQL Update Delay Time
- SQL Update Delay Percent
- Total Time
- User-defined Delay Percent
- User-defined Delay Time
- Workload Full Name
- Workload Type

For detailed information about these attributes and their values, refer to the online Help for OMEGAMON XE for WebSphere Application Server for OS/390.

Introducing Selected Workload Information

Overview

The Selected Workload - Current Interval and the Selected Workload - History workspaces report additional detailed data about a particular instrumented workload.

Accessing the Selected Workload - Current Interval workspace

Use these steps to access the Selected Workload - Current Interval workspace.

1. Expand the Navigator.
2. Expand the WebSphere Application Server OS/390 node to access its nested items.
3. Select the J2EE Server item and expand its Navigator tree.
4. Select the application server you want and expand its Navigator tree.
5. Choose the All Workloads selection in the Navigator.
6. Right-click on the row containing the workload you are interested in the All Workloads - Current Interval table.
7. Choose **Link To > Selected Workload - Current Interval** from the pop-up menus.

Inside the Selected Workload - Current Interval workspace

The predefined Selected Workload - Current Interval workspace contains

- a pie chart that shows delay times in milliseconds for the workload during the interval
- a Selected Workload - Current Interval table view that displays information about contention, delays, times and the names of the monitor object class, waiting class, waiting method, and waiting method signature

Use

The Selected Workload - Current and Selected Workload - History workspaces report detailed response time information to show how the workload spent its time. Their charts let you quickly identify problem areas where delays occurred. The table views also show the major and minor names of the delays, the number of occurrences, and rate of occurrence for specific methods.

Important performance measurements in these workspaces are the average execution time the workload spent in each delay, the percentage of total workload time that was consumed by the delays, and the CPU time consumed by each delay.

Historical data

If you want to view this data over a time span, you can see collected data in the Selected Workload - History workspace. The default history workspace contains a bar chart that shows the average response time for each type of delay during the last four hours. It also contains a Selected Workload - History table view.

This historical data is collected over a measured time span. Select a row in the Selected Workload - Current Interval table view, and right-click to open the pop-up menu and link to the historical view. See the online Help for CandleNet Portal for detailed explanation of historical reporting.

Information presented in the Selected Workload table view

This is a list of the Workload Degradation Detail for WAS OS/390 attributes that report data in the Selected Workload table view.

- Average Time
- Class Name
- CPU Active Percent
- CPU Active Samples
- CPU Contention Percent
- CPU Contention Samples
- CPU Samples
- CPU Time
- CPU Wait Percent
- CPU Wait Samples
- Delay Full Name
- Delay Major Name
- Delay Minor Name
- Delay Percent
- Delay Type
- Interval Time
- Max Time
- Method Name
- Number of Occurrences
- Node Name
- Origin Node
- Process ID
- Sample Date and Time
- Server Instance Name
- Server Name
- Total Time
- Workload Type

For detailed information about these attributes and their values, refer to the online Help for OMEGAMON XE for WebSphere Application Server for OS/390.

Introducing Selected Delay Information

Overview

The Selected Delay - Current Interval and the Selected Delay - History workspaces report details about specific delays that were encountered by a selected workload.

Accessing the Selected Delays workspace

Use these steps to access the Selected Delay - Current workspace.

1. Expand the Navigator.
2. Expand the WebSphere Application Server OS/390 node to access its nested items.
3. Select the J2EE Server item and expand its Navigator tree.
4. Select the application server you want and expand its Navigator tree.
5. Choose the All Workloads selection in the Navigator.
6. Click the link beside a row in the All Workloads table for the workload you are interested in and choose Selected Delay - Current Interval or choose the workspace from the **View > Workspaces** menu.

Inside the Selected Delay - Current workspace

The predefined Selected Delay - Current workspace contains

- a bar chart that shows graphically all of the delays that impacted this workload. The length of each bar indicates the average amount of time, in milliseconds, that the workload was delayed, per execution of the workload.
- A Selected Workload Delays - Current Interval table view that shows more detailed information about each delay which impacted the selected workload.

Use

The Selected Delay - Current and Selected Delays - History table views contains one row of detailed data for each specific delay encountered for the selected workload. Important performance measurements in this workspace are the average execution time the workload spent in each delay, the percentage of total workload time that was consumed by each delay, and the CPU time consumed by each delay for this workload.

Historical data

If you want to view this data over a time span, you can see collected data in the Selected Delays - History workspace. In addition to the Selected Delay - History table view, the default history workspace contains two plot charts that show data collected during the last four hours. One shows the average response times for the selected delay; the other shows occurrences of the selected delay.

This historical data is collected over a measured time span. Select a row in the Selected Delay - Current table view, and right-click to open the pop-up menu and link to the historical view. See the online Help for CandleNet Portal for detailed explanation of historical reporting.

Information presented in the selected delays table views

Both the Selected Delay - Current and Selected Delay - Historical table views display data provided by the Workload Delay Details for WAS OS/390 attributes shown on the following page.

This is a list of the Workload Degradation Detail for WAS OS/390 attributes that report data in the Selected Workload table view.

- Average Time
- Class Name
- CPU Active Percent
- CPU Active Samples
- CPU Contention Percent
- CPU Contention Samples
- CPU Samples
- CPU Time
- CPU Wait Percent
- CPU Wait Samples
- Delay Full Name
- Delay Major Name
- Delay Minor Name
- Delay Percent
- Delay Type
- Interval Time
- Max Time
- Method Name
- Number of Occurrences
- Node Name
- Origin Node
- Process ID
- Sample Date and Time
- Server Instance Name
- Server Name
- Total Time
- Workload Type

For detailed information about these attributes and their values, refer to the online Help for OMEGAMON XE for WebSphere Application Server for OS/390.

Introducing the Ten Most Frequently Used Workloads

Overview

The Ten Most Frequently Used Workloads workspace provides a quick way to identify which workloads are being used the most.

Accessing the Ten Most Frequently Used Workloads workspace

Use these steps to access the Ten Most Frequently Used Workloads workspace.

1. Expand the Navigator.
2. Expand the WebSphere Application Server OS/390 to access its nested items.
3. Select the J2EE Server item and expand its Navigator tree.
4. Select the application server you want and expand its Navigator tree.
5. Choose the All Workloads selection in the Navigator.
6. Right click on the row containing the workload you are interested in the All Workloads - Current Interval table.
7. Choose **Link To > Ten Most Frequently Used Workloads** from the pop-up menus.

Inside the Ten Most Frequently Used Workloads workspace

The predefined Ten Most Frequently Used Workloads workspace contains

- a bar chart that shows the delay response times for the 10 most frequently used workloads
- a bar chart that shows the number of occurrences for each of the 10 most frequently used workloads
- a Ten Most Frequently Used Workloads table view that displays the metrics for each workload

Use

The Ten Most Frequently Used Workloads workspace lets you quickly identify which workloads are being used the most, their response times, and the frequency of occurrence. The table view contains a row of data for each of these workloads.

Information presented in the Ten Most Frequently Used Workloads table view

The Ten Most Frequently Used Workloads table view displays data provided by the Workload Degradation Summary for WAS OS/390 attributes. For a list of these attributes, see page 168.

Introducing Longest Running Workloads Information

Overview

Both the Longest Running Workloads and the Selected Longest Running Workload - History workspaces report detailed information about delay times for the longest running workloads. These workloads track individual transaction executions that exceed thresholds for a response time that you specify. The workspaces report details about these exceptions.

Collecting data for the longest running workloads

In addition to setting up the agent to collect workload information, you also need to define thresholds for workloads that will trigger the reporting of exception transactions. To do this, use the WAS390 Set Workload Exception Thresholds take action command.

Refer to the take action command Help for the OMEGAMON XE agent for details about using this command. The workspaces for the longest running workloads will not be able to display data if you do not issue this command. It is also possible to define the thresholds for exception workloads in the KWWXML member.

Accessing the Longest Running Workloads workspace

Use these steps to access the Longest Running Workloads workspace.

1. Expand the Navigator.
2. Expand the WebSphere Application Server OS/390 node to access its nested items.
3. Select the J2EE Server item and expand its Navigator tree.
4. Select the application server you want and expand its Navigator tree.
5. Choose the Longest Running Workloads selection in the Navigator.

Inside the Longest Running Workloads workspace

The predefined Longest Running Workloads workspace contains

- a bar chart that shows the delay times in milliseconds for the longest running workloads. Each bar in this chart represents an individual execution of a workload.
- a Longest Running Workloads - Current Interval table view that shows the type of workloads, their response times, and delay information for them. Each row in this table view represents an individual execution of a workload.

Use

Both the Longest Running Workloads and Selected Longest Running Workload - History workspace provide breakdowns of where these workloads are spending their time. The current table view contains one row of detailed data for each specific long-running workload. These workloads are determined by the filtering criteria you set with the take action command. Important data in this table view include the user ID and IP address, which tell you which browser initiated the transaction; the response time; and the percentage of time that the workload spent performing the activity.

Note: *You will only see a user ID for the workload if the WebSphere Application Server is aware of the user ID.*

Historical data

If you want to view this data over a time span, you can see collected data in the Selected Longest Running Workload - History workspace. In addition to the Selected Longest Running Workload - History table view, the default history workspace contains two plot charts that show data collected during the last four hours. One shows the average response times for the selected delay, the other occurrences of the selected delay.

This historical data is collected over a measured time span. Select a row in the Longest Running Workloads table view, and right-click to open the pop-up menu and link to the historical view. See the online Help for CandleNet Portal for detailed explanation of historical reporting.

Information presented in the selected delays table views

Both the Longest Running Workloads and Selected Longest Running Workload - Historical table views display data provided by the Workload Exception for WAS OS/390 attributes shown on the following page.

This is a list of the Workload Exception for WAS OS/390 attributes.

- Class Name
- EJB Home Delay Percent
- EJB Home Delay Time
- EJB Method Delay Percent
- EJB Method Delay Time
- EJB Remote Delay Percent
- EJB Remote Delay Time
- Interval Time
- IP Address
- JMS Delay Time
- JMS Delay Percent
- JNDI Delay Time
- JNDI Delay Percent
- JTA Delay Time
- JTA Delay Percent
- Method Name
- Miscellaneous Delay Percent
- Miscellaneous Delay Time
- Network Socket Delay Percent
- Network Socket Delay Time
- Node Name
- Number EJB Home Delays
- Number EJB Method Delays
- Number EJB Remote Delays
- Number JMS Delays
- Number JNDI Delays
- Number JTA Delays
- Number Miscellaneous Delays
- Number Network Socket Delays
- Number Servlet Delays
- Number SQL Connection Delays
- Number SQL Query Delays
- Number SQL Update Delays
- Number User-defined Delays
- Origin Node
- Process ID
- Response Time
- Sample Date and Time
- Server Instance Name
- Server Name
- Servlet Delay Percent
- Servlet Delay Time
- SQL Connection Delay Percent
- SQL Connection Delay Time
- SQL Query Delay Percent
- SQL Query Delay Time
- SQL Update Delay Percent
- SQL Update Delay Time
- User-defined Delay Percent
- User-defined Delay Time
- User ID
- Workload Full Name
- Workload Type

For detailed information about these attributes and their values, refer to the online Help for OMEGAMON XE for WebSphere Application Server for OS/390.

Introducing Data Sources Information

Introduction

This chapter discusses the workspaces for data source information that provides information about J2EE Datasources through logical connections to database systems.

Accessing the Datasources workspace

Use these steps to access the Datasources workspace.

1. Expand the Navigator.
2. Expand the WebSphere Application Server OS/390 node to access its nested items.
3. Select the J2EE Server item and expand its Navigator tree.
4. Select the application server you want and expand its Navigator tree.
5. Choose the Datasources selection in the Navigator.

Contents of the Datasources workspace

The predefined Datasources workspace contains

- a bar chart that displays the average time used for waiting for database database connections, processing SQL queries, and making updates
- a Datasources - Current Interval table view that displays details about each data source

Use

The Datasource and Selected Datasource - History workspaces report information response times for accessing database information for the current interval and for an extended period of time in a history view. The table contains one row of data for each data source and identifies the database product and version. It also gives you such details as the rates for connections, queries, and updates, as well as the processing times for these tasks.

Historical data

You can see collected data collected over a time span in the Selected Datasources - History workspace. The default history workspace contains a plot chart that shows the average wait to obtain information from a database and the SQL processing time during the last four hours. It also contains a Selected Datasource - History table view.

This historical data is collected over a measured time span. Select a row in the Datasources table view, and right-click to open the pop-up menu and link to the historical view. See the online Help for CandleNet Portal for detailed explanation of historical reporting.

Information presented in the Datasources table views

The Datasource Summary - Current Interval and Selected Queue - History table views display data provided by the Datasource Detail for WAS OS/390 attributes report one row for each data source. The following is a list of these attributes

- Average Connection Wait Time
- Average Query Processing Time
- Average Update Processing Time
- Connection Rate
- Database Product
- Database Product Version
- Datasource Name
- Interval Time
- Max Connection Wait Time
- Node Name
- Number of Connections
- Number of Queries
- Number of Updates
- Origin Node
- Process ID
- Query Rate
- Sample Date and Time
- Server Instance Name
- Server Name
- Total Connection Wait Time
- Total Query Processing Time
- Total Update Processing Time
- Total Wait Time
- Update Rate

For detailed information about these attributes and their values, refer to the online Help for OMEGAMON XE for WebSphere Application Server for OS/390.

Introducing HTTP Session Information

Introduction

This chapter discusses the HTTP Sessions workspace that provides performance information about current HTTP sessions.

Accessing the HTTP Sessions workspace

Use these steps to access the HTTP Sessions workspace.

1. Expand the Navigator.
2. Expand the WebSphere Application Server OS/390 node to access its nested items.
3. Select the J2EE Server item and expand its Navigator tree.
4. Select the application server you want and expand its Navigator tree.
5. Choose the HTTP Sessions selection in the Navigator.

Contents of the HTTP Sessions workspace

The predefined HTTP Sessions workspace contains

- a plot chart that shows the number of HTTP seasons during the last four hours
- a HTTP Sessions - Current Interval table view that displays details about each currently active session

Use

The HTTP Sessions workspace reports information associated with HTTP sessions. The table contains one row of data for each current session. This information identifies the session ID, the session creator, and the name of the web application associated with the session. It also includes the IP address for the browser and associated attributes.

Information presented in the HTTP Sessions table views

The HTTP Sessions - Current Interval table view displays data provided by the HTTP Session Detail for WAS OS/390 attributes. The following is a list of these attributes.

- Associated Attributes
- Class Name
- Create Date and Time
- Creating IP Address
- Creating User
- Interval Time
- Last Accessed Time
- Maximum Inactive Interval
- Method Name
- Node Name
- Origin Node
- Process ID
- Sample Date and Time
- Server Instance Name
- Server Name
- Session ID
- Web Application Name
- Workload Type

For detailed information about these attributes and their values, refer to the online Help for OMEGAMON XE for WebSphere Application Server for OS/390.

Introducing Messaging and Queue Information

Introduction

This chapter discusses the messaging workspaces that provide information about queues being used by the Java Messaging Service (JMS) interface and the MQSeries Access middleware messaging application.

Accessing the JMS Summary workspace

Use these steps to access the JMS Summary workspace.

1. Expand the Navigator.
2. Expand the WebSphere Application Server OS/390 node to access its nested items.
3. Select the J2EE Server item and expand its Navigator tree.
4. If your installation has multiple nodes, select the system you want to examine and expand its Navigator tree.
5. Select the J2EE Server item and expand its Navigator tree.
6. Select the application server you want and expand its Navigator tree.
7. Choose the JMS Summary selection in the Navigator.

Contents of the JMS Summary workspace

The predefined JMS Summary workspace contains

- a bar chart that shows the average response times for message browse, put, and get activities for each MQSeries queue accessed during the interval
- a JMS Summary - Current Interval table view that displays details about each MQSeries queue accessed during the interval

Use

This workspace reports JMS information about how WebSphere applications are using MQSeries. It identifies the queues and queue managers and also contains the rates and specific times for browse, put, and get activities.

Historical data for a selected queue

If you want to view this for a specific queue over a time span, you can see historical data in the Selected Queue - History workspace. The default history workspace contains a plot chart that shows the average response times for browse, put, and get activities for the MQSeries Access queue during the last four hours. It also contains a Selected Queue - History table view.

This historical data is collected over a measured time span. Select a row in the Datasources table view, and right-click to open the pop-up menu and link to the historical view. See the online Help for CandleNet Portal for detailed explanation of historical reporting.

Information presented in the JMS Summary and Selected Queue - History table views

The MQSeries Access for WAS OS/390 attributes report data in the JMS Summary - Current Interval and Selected Queue - History table views. The following is a list of these attributes.

- Average Browse Time
- Average Get Time
- Average Put Time
- Browse Rate
- Full Name
- Get Rate
- Interval Time
- Node Name
- Number of Browse Gets
- Number of Gets
- Number of Puts
- Origin Node
- Process ID
- Put Rate
- Queue Manager Name
- Queue Name
- Sample Date and Time
- Server Instance Name
- Server Name
- Total Browse Time
- Total Get Time
- Total Put Time
- Total Time

For detailed information about these attributes and their values, refer to the online Help for OMEGAMON XE for WebSphere Application Server for OS/390.

Introducing Application Trace File Information

Overview

OMEGAMON XE for WebSphere Application Server for OS/390 includes an application trace facility that enables you to trace the flow for one or more instrumented workloads. You start this facility from within CandleNet Portal.

Both the Application Trace Files and the Application Trace workspaces report tracing data that track the flow within the workloads you have instrumented. The following topics provide background about collecting this data and discusses the type of tracing information that these workspaces provide.

Prerequisites

The collection of application trace data assumes you have met the following three prerequisites.

1. Changed the agent's default configuration

Use the DefineAppTraceDirectory parameter in the KWWXML member to provide a name for the application trace directory in which you want to put the collected data.

2. Prepared your system for workload analysis

Complete the following tasks to provide support for workload data collection.

A. Used the parameters and keywords in the kwwiuser.xml file to define instrumentation for the workloads you want to track.

B. Executed the Candle instrumentation utility.

C. Changed the WAS for z/OS and OS/390 configuration to support workload Candle's workload analysis

3. Initiated workload data collection

Use the WAS390 Dynamic Workload Analysis take action command to activate the collection of workload data.

Refer to Chapter 2 "Configuring OMEGAMON XE for WebSphere Application Server for OS/390" on page 39 for configuration and instrumentation details and the agent's take action commands for information about using these commands.

Collecting Application Trace Data

Overview

This topic discusses the take action commands that you use to control the collection of application trace data and explains the steps you need to perform to begin a trace.

Take action commands for application traces

You control tracing from within CandleNet Portal by using take action commands that let you turn tracing on and off and delete unwanted traces. You can limit traces by the number of entries or time. The take action commands that pertain to application tracing are

- WAS390 Start Application Tracing

  Use this command to create a trace file and begin recording to it. The tracing stops and this file is closed when the amount of time for the trace or the trace entries specified on the command. The trace will show each entry and exit from an instrumented Java method. The WAS390 Stop Application Tracing take action command can also be used to stop the trace.

- WAS390 Stop Application Tracing

  Use this command to stop an application trace in order to view the collected in the application trace workspaces. Once the trace has been turned off, the OMEGAMON XE agent can read the trace file and display the entries in a table view.

- WAS390 Delete Application Tracing Files

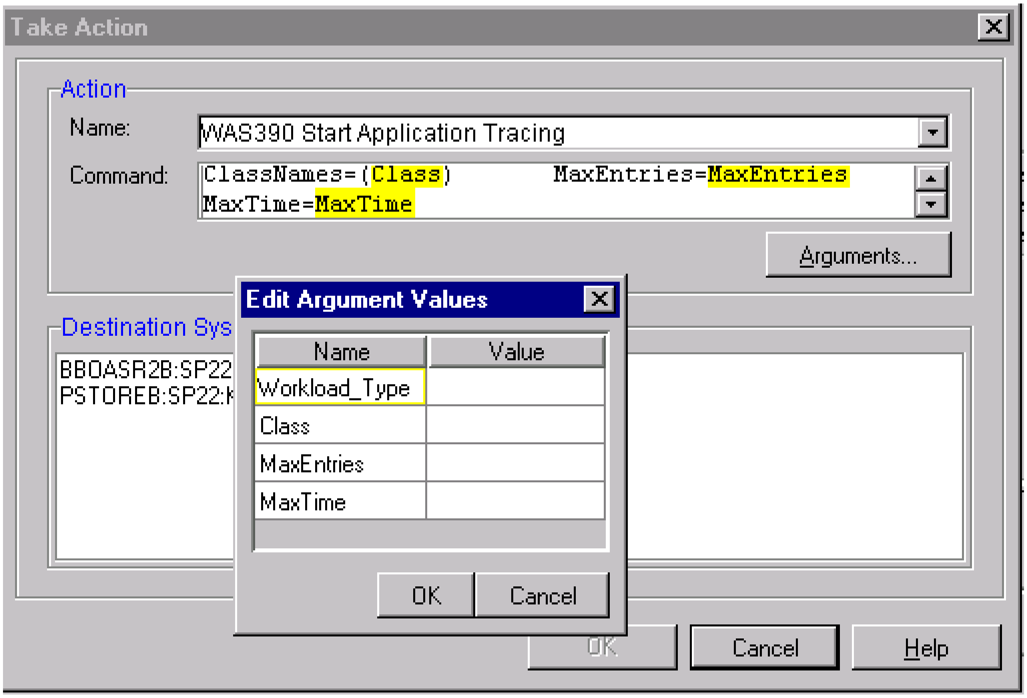
  Use this command to delete an unwanted application trace file.

For additional details about using these commands, see the take action command Help for the OMEGAMON XE agent.

Starting an application trace

Perform these steps to start an application trace on an instrumented workload.

1. In the CandleNet Portal Navigator, select any J2EE workspace that contains workload information you want to trace, for example, Databases. Right-click and choose **Take Action** from the pop-up menu.
2. In the Take Action dialog, scroll the commands in the Name field and choose **WAS390 START Application Tracing** to display the following dialog.



Name: WAS390 Start Application Tracing

Command: ClassNames=(Class) MaxEntries=MaxEntries MaxTime=MaxTime

Name	Value
Workload_Type	
Class	
MaxEntries	
MaxTime	

BBOASR2B:SP22
PSTOREB:SP22:K

3. Complete the Value fields described below.

Table 7. WAS390 Start Application Tracing Command Values

Parameter Name	Explanation
Workload_Type	Provide one of the following values to specify type of workload you want to trace: **Servlet**, **EJB**, or **Both**.
Class	Provide the name of the instrumented class that you want to trace. You can provide multiple class masks, separated by commas or semi-colons. For example, a mask of **ClassName=*** specifies that trace data will be collected for all classes that match the workload type.

Table 7. WAS390 Start Application Tracing Command Values (continued)

Parameter Name	Explanation
MaxEntries	Specify the maximum number of entries you want the trace to collect.
MaxTime	Specify the maximum length of time in seconds that you want the trace to run.

Introducing the Application Trace Files Workspace

Overview

The Application Trace Files workspace reports trace information about each file in the trace directory for OMEGAMON XE for WebSphere Application Server for OS/390. These application trace files are those traces that you have run to monitor activity and measure performance.

Accessing the Application Trace Files workspace

To access the Application Trace Files workspace for a system,

1. Expand the Navigator.
2. Expand the WebSphere Application Server OS/390 node to access its nested items.
3. Expand the system you want to examine to access its nested items.
4. Choose the Application Trace Files selection in the Navigator.

Inside the Application Trace Files workspace

The default Application Trace Files workspace contains an Enterprise Event Console and an Application Trace Files table view. The table view lists all the workload traces for application server instances. It contains one row of data for each available trace file.

Use

This workspace provides an immediate view of all traces on instrumented workloads that you have initiated. The Application Trace Files table view provides details about any traces that the OMEGAMON XE agent performed for the application server and the start time of the trace.

From within the table view, you can select a particular trace to link to its Application Trace workspace, which then provides details about the specific trace.

Information presented in the Application Trace Files table view

The Application Trace Files table view provides data reported by the Application Trace File for WAS OS/390 attributes. The following is a list of these attributes.

- Class Name
- Interval Time
- Method Name
- Node Name
- Origin Node
- Process ID
- Sample Data and Time
- Server Instance Name
- Server Name
- Start Time
- Trace ID
- Workload Type

For detailed information about these attributes and their values, refer to the online Help for OMEGAMON XE for WebSphere Application Server for OS/390.

Introducing the Application Trace Workspace

Overview

The Application Trace workspace reports the detailed trace entries contained within a selected trace file. Each entry in the trace file table view represents an event (method call, method return, workload start, workload end, or thrown exception) in the lifecycle of a workload which was selected for application tracing.

Accessing the Application Trace workspace

To access the Application Trace workspace for a system,

1. Expand the Navigator.
2. Expand the WebSphere Application Server OS/390 node to access its nested items.
3. Choose the Application Trace Files selection in the Navigator.
4. Select a row for a trace that you want to explore and right-click to open the pop-up menu. Choose the Application Trace selection.

Inside the Application Trace workspace

The default Application Trace workspace contains an Enterprise Event Console and an Application Trace table view. The table view shows all the events that occurred during this application trace.

Note: *You must turn off the trace before you can see details about the data that was collected for the application trace.*

Use

This workspace provides detailed information that shows the class stack level, the method that was called, and what type of event occurred. For example, the event type tells you whether an exception, method, or workload was reported, and the class and method information tells you what was called.

Information presented in the Application Trace table view

The Application Trace table view provides data reported by the Application Trace for WAS OS/390 attributes. The following is a list of these attributes.

- Call Level
- Class
- Event Type
- IP Address
- Interval Time
- Method
- Object
- Origin Node
- Parameters
- Process ID
- Sample Data and Time
- Server Instance Name
- Server Name
- Thread
- Time
- Trace ID
- User ID

For detailed information about these attributes and their values, refer to the online Help for OMEGAMON XE for WebSphere Application Server for OS/390.

10

JVM Heap Usage

Overview

This chapter provides instructions for accessing the JVM Heap Usage workspace and discusses its contents.

Chapter contents

Introducing JVM Heap Usage Information

Introduction

This chapter discusses the JVM Heap Usage workspace that provides information from the profiler about the use of heap storage within the Java Virtual Machine (JVM) that is hosting an application server.

Accessing the JVM Heap Usage workspace

Use these steps to access the JVM Heap Usage workspace.

1. Expand the Navigator.
2. Expand the WebSphere Application Server OS/390 node to access its nested items.
3. Expand a *server_node_name* you want to examine to access its nested items.
4. Select the J2EE Server or MOFW Server item and expand its Navigator tree.
5. Select the application server you want to inspect and expand its Navigator tree.
6. Choose the JVM Heap Usage selection in the Navigator.

Contents of the JVM Heap Usage workspace

The predefined JVM Heap Usage workspace contains

- a bar chart that shows which Java classes are consuming the most space on the heap, in terms of the number and size of the class instantiations.
- a bar chart that shows the 10 most frequently allocated classes, in terms of number of class instantiations per second.
- a JVM Heap Usage table view that displays additional information regarding class names, process IDs, objects, bytes used, and allocation requests

Move the cursor over different areas of the bar charts to display flyover details about the items in a chart.

JVM Heap Usage Table View

Use

This workspace reports information associated with heap usage and enables you to determine, by Java class, how the heap is being used. It provides data from the Candle profiler about the use of heap storage within the Java Virtual Machines that are hosting an application server.

Certain data collection is turned off by default because of high overhead. If you want to report this data in the workspace, then activate the JVM profiler or the set the instrumentation level for certain attributes as needed. See "Activating the JVM Profiler" on page 66 for information about activating the profiler and the online Help for setting instrumentation levels for attributes.

Information presented in the JVM Heap Usage table view

In the table view, the Java Heap Usage for WAS OS/390 attributes report one row of data for each class defined in the JVM running an application server. This is a listing of the JVM Heap Usage attributes reported in the Java Heap Usage table view.

- Allocation Rate
- Allocation Requests
- Class Name
- Interval Time
- Node Name
- Number of Bytes
- Number of Objects
- Objects Freed
- Objects Moved
- Origin Node
- Process ID
- Sample Date and Time
- Server Instance Name

For detailed information about these attributes and their values, refer to the online Help for OMEGAMON XE for WebSphere Application Server for OS/390.

12

Application Server Error Log

Overview

This chapter discusses the Application Server Error Log workspace. This workspace provides a quick view of errors and exceptions reported for OMEGAMON XE for WebSphere Application Server for OS/390.

Chapter contents

Introducing Application Server Error Log Information

Overview

OMEGAMON XE for WebSphere Application Server for OS/390 scans the OMEGAMON XE for WebSphere Application Server for OS/390 logstreams to detect errors and exception conditions within the servers. This chapter discusses this reporting feature.These errors or exception conditions may be associated with a specific application server or may be associated with multiple application servers.

Accessing the Application Server Error Log workspace for multiple application servers

Use these steps to access the Application Server Error Log workspace for a multiple application server.

1. Expand the Navigator.
2. Expand the WebSphere Application Server OS/390 node to access its nested items.
3. Expand a *server_node_name* you want to examine to access its nested items.
4. Select the J2EE Server or MOFW Server item and expand its Navigator tree.
5. Choose the Application Server Error Log selection in the Navigator.

Accessing the Application Server Error Log workspace for a single application server

Use these steps to access the Application Server Error Log workspace for a single application server.

1. Perform steps 1–4 above.
2. Select the *server_subnode_name* you want to inspect and expand its Navigator tree.
3. Select the application server you want to inspect and expand its Navigator tree.
4. Choose the Application Server Error Log selection in the Navigator.

Inside the Application Server Error Log workspace

The predefined Application Server Error Log workspace contains

- an Enterprise Event Console that displays changes in the status of situations that have been assigned to the console view
- an Application Server Error Log table view that displays message text and related information from the Application Server Error Log attributes

Data history

If you want to view this data over a particular time span, you can see collected data in the Application Server Error Log History workspace. This history workspace also contains an Enterprise Event Console and an Application Server Error Log History table view that displays messages and related information about application server errors

This historical data is collected over a measured time span. See the online Help for CandleNet Portal for detailed explanation of historical reporting.

To access the history workspace, use any of the following procedures.

- **Select View > Workspace** from the menu bar in the Application Server Error Log workspace and choose the history workspace from a pull-down menu.
- Select and right-click an application server instance in the Navigator choose the history workspace from the pop-up menu.
- Double click a row in the Application Server Instance table view to link to the history workspace.

Application Server Error Log Table View

Use

The Applications Server Error Log table view provides an immediate view of errors and exception conditions reported by an error logstream. Use it for quick access to summary information for problem tracking purposes.

Information presented in the Application Server Error Log table view

The Application Server Error Log table view displays data reported by the Application Server Error Log attributes. This view enables you to examine error and exception data collected over an extended time span that you specify.

This table view displays data provided by the Application Server Error Log attributes. The following is a list of attributes reported in the Application Server Error Log table view.

- Error Date and Time
- Host Name
- Job Address Space Identifier
- Job Name
- Message Identifier
- Message Origin
- Message Text
- Origin Node
- Process Identifier
- Server Instance Name
- Thread Identifier
- Identifier

For detailed information about these attributes and their values, refer to the online Help for OMEGAMON XE for WebSphere Application Server for OS/390.

11

JVM Garbage Collector Activity

Overview

This chapter provides instructions for accessing the JVM Garbage Collection Activity workspace and discusses its contents.

Chapter contents

Introducing JVM Garbage Collection Information

Introduction

This chapter discusses the JVM Garbage Collection Activity workspace. This workspace provides information about the garbage collector in the Java Virtual Machines which are hosting an application server. This information is reported by the Candle profiler.

Accessing the JVM Garbage Collector Activity workspace

Use these steps to access the JVM Garbage Collector Activity workspace.

1. Expand the Navigator.
2. Expand the WebSphere Application Server OS/390 node to access its nested items.
3. Expand a *server_node_name* you want to examine to access its nested items.
4. Select the J2EE Server MOFW Server item and expand its Navigator tree.
5. Select the application server you want to inspect and expand its Navigator tree.
6. Select the application server you want and expand its Navigator tree.
7. Choose the JVM Garbage Collector Activity selection in the Navigator.

Inside the JVM Garbage Collector Activity workspace

This workspace provides information from the JVM profiler about the garbage collector activity for the Java Virtual Machine that is hosting an application server.

The predefined JVM Garbage Collector Activity workspace contains

- a bar chart that shows the number of garbage collections during the interval
- a bar chart that shows the percent of CPU that was used by garbage collection activity during the interval
- a JVM Garbage Collector Activity table view that displays detailed information such as the objects freed, CPU usage and so forth.

Note: *Move the cursor over different areas of a bar chart to display flyover details about the items in a chart.*

JVM Garbage Collector Activity Table View

Use

The JVM Garbage Collector Activity workspace reports information about the number of times the collector ran during the interval, the CPU used by the collector, and the number of objects freed by the collector. Use the JVM Garbage Collector Activity workspace to monitor garbage collection performance and problems.

Data collection by the JVM profiler is turned off by default because of high overhead. If you want to report this data in the workspace, then activate the Candle profiler. See "Activating the JVM Profiler" on page 66 for this information.

Information presented in the JVM Garbage Collector Activity table view

The JVM Garbage Collector Activity table view shows detailed garbage collection information. The following is a list of JVM Garbage Collector Activity for WAS OS/390 attributes that are reported in this table view.

- CPU Percent
- Interval Time
- Node Name
- Objects Freed
- Objects Moved
- Origin Node
- Process ID
- Sample Date and Time
- Server Instance Name
- Times Run
- Total CPU Used

For detailed information about these attributes and their values, refer to the online Help for OMEGAMON XE for WebSphere Application Server for OS/390.

A

OMEGAMON XE for WebSphere Application Server for OS/390 Messages

Introduction

This appendix contains the KWE-prefixed messages that are produced by the OMEGAMON XE agent for WebSphere Application Server. It also contains the KWEWA- prefixed messages that issued within the WebSphere Application Server addresss space. The KWEWA messages begin on page 230.

KWE messages

The following messages cover a numbering range of KWE0001–KWE0079E and are listed alphabetically.

KWE0001I **OMEGAMON XE for WebSphere Application Server startup complete**

Explanation: The OMEGAMON XE monitoring agent has successfully completed initialization.

User Response: None. This an informational message only.

KWE0002E *cccc* **exception caught in** *module_name function_name***.**

Explanation: A program exception has occurred in the OMEGAMON XE monitoring agent. The variable in the message indicate the type of exception, the module in which the exception occurred, and the function involved.

User Response: Collect the agent log and contact Candle Customer Support.

KWE0003E **Environmental error** *error_type* **caught in** *function_name* **RC =** *return_code*

Explanation: An environmental error has occurred in the OMEGAMON XE monitoring agent. The variables in the message indicate the type of error, the name of the function where the error was detected, and the return code.

User Response: Collect the agent log and contact Candle Customer Support.

KWE0004E **Exception *cccc* caught.**

Explanation: This message is produced after an exception condition is caught in the Windows environment. This variable in the message indicates the type of exception that was caught.

User Response: Collect the agent log and contact Candle Customer Support.

KWE0005E **Exception address *xxxx*.**

Explanation: This message is produced after an exception condition has occurred in the Windows environment. The *xxxx* variable indicates the address where the exception took place.

User Response: Collect the agent log and contact Candle Customer Support.

KWE0007I **New application server *servername* discovered.**

Explanation: The OMEGAMON XE monitoring agent has discovered a defined application server in the WebSphere confutation.

User Response: Collect the agent log and contact Candle Customer Support.

KWE0008I **Background EPM monitor started for application server *servername*.**

Explanation: The OMEGAMON XE monitoring agent has started Enterprise Performing Monitoring (EPM) for the application server identified by the *servername* variable. OMEGAMON XE for WebSphere Application Server for OS/390 will now collect EPM performance information on this application server.

User Response: None. This an informational message only.

KWE0009I **Background EPM monitor stopping for application server *servername*.**

Explanation: OMEGAMON XE monitoring agent has stopped Enterprise Performing Monitoring (EPM) for the application server indicated by the *servername* variable. OMEGAMON XE will no longer collect EPM performance information about this application server.

User Response: None. This an informational message only.

KWE0010I **Background profiler monitor started for application server *servername*.**

Explanation: The OMEGAMON XE monitoring agent has started profiler monitoring for the application server indicated by the variable, *servername*.

OMEGAMON XE for WAS will now collect profiler performance information on this application server.
User Response: None. This an informational message only.

KWE0011I **Background profiler monitor stopping for application server *servername*.**
Explanation: The OMEGAMON XE for WAS monitoring agent has stopped profile monitoring for the application server indicated by the *servername* variable. One of the following conditions has occurred.

- Profiler data collection has ended normally.

  or

- There was a problem with the profiler for the application server, in which case, there should be a prior message indicating the cause of the problem.

OMEGAMON XE will no longer collect profiler performance information on the application server.

User Response: None. This an informational message only.

KWE0012I **Profiler session started for application server *servername*.**
Explanation: The OMEGAMON XE for WAS monitoring agent has made contact with the profiler running on the application server indicated by the *servername* variable.
User Response: None. This an informational message only.

KWE0013I **Started collecting profiler data for application server *servername*.**
Explanation: The OMEGAMON XE WAS monitoring agent has initiated profile performance data collection for the application server indicated by the *servername* variable.
User Response: None. This an informational message only.

KWE0014I **Stopping collecting profiler data for application server *servername*.**
Explanation: The OMEGAMON XE for WAS monitoring agent has stopped collecting profiler performance data for the application server indicated by the variable *servername*. One of the following conditions has occurred.

- Profiler performance data collection has ended normally.
- There was a problem with the profiler for the application server, in which case, there should be a prior message indicating the cause.

OMEGAMON XE will no longer collect profiler performance information on the application server.

User Response: None. This an informational message only.

KWE0015E **Bad return code *xxxxxxxx* received on socket interface for application server *servername*.**

Explanation: An error occurred on the interface to the profiler running on the application server indicated by the *servername* variable. The error indicated by the return code can be one of the following conditions.

10	Network_Subsystem_Down
11	Socket_Not_Connected
12	Socket_Shutdown
13	Virtual_Circuit_Terminated
14	Virtual_Circuit_Reset
15	Unknown_Socket_Error
17	Socket_Timeout
22	Permission_To_Create_Socket_Denied
23	Descriptor_Table_Full
24	Insufficient_Memory_For_Socket
25	Insufficient_Streams_Available
26	Unsupported_Protocol
27	Interrupted_By_Signal
28	Message_Too_Long
29	Socket_IO_Error

User Response: Attempt to determine why the error condition specified in the message occurred. This error can occur if the application server was stopped.

KWE0016E **Socket interface for application server *servername* shut down.**

Explanation: The connection between the OMEGAMON XE for WAS monitoring agent and the profiler running on the application server indicated by the *servername* variable was ended. An error is probably the likely cause.

User Response: Look for the KWE0015E message, which should have preceded this one. Follow the user response instructions for the KWE0015E message.

KWE0017I **Application server *servername* is now active.**

Explanation: OMEGAMON XE for WAS has detected the fact that the application server identified by the *servername* variable has become active.
User Response: None. This an informational message only.

KWE0018I **Application server *servername* is no longer active.**

Explanation: The OMEGAMON XE for WAS monitoring agent has detected that the application server identified by the *servername* variable is no longer active.
User Response: None. This an informational message only.

KWE0019E **Unable to load Java Virtual Machine shared library.**

Explanation: The OMEGAMON XE for WAS monitoring agent was unable to load the Java Virtual Machine shared library. The library name is jvm.dll on Windows, libjvm.a on AIX, and libjvm.so on Solaris.
User Response: Ensure that the WASAppServerRoot parameter in the kwe.xml file correctly identifies the root directory where the Websphere Application Server is installed.

KWE0020E **Unable to locate JNI_CreateJavaVM within Java Virtual Machine shared library.**

Explanation: The OMEGAMON XE for WAS monitoring agent was unable to locate the JNI_CreateJavaVM entry point in the Java Virtual Machine shared library. The library name is jvm.dll on Windows, libjvm.a on AIX, and libjvm.so on Solaris.
User Response: Contact Candle Customer Support.

KWE0021E **Unable to create Java Virtual Machine. RC = *return_code*.**

Explanation: The OMEGAMON XE for WAS monitoring agent was unable to create the Java Virtual Machine. The *return_code* variable is the code received from the JNI_CreateJavaVM function in the JVM shared library.
User Response: Contact Candle Customer Support.

KWE0022E **Unable to locate KWEEPMClient class.**

Explanation: The OMEGAMON XE for WAS monitoring agent was unable to locate the KWEEPMClient Java class.
User Response: Ensure that the lwepmclient.jar file (for a WAS 3.5.x environment) or the kwepmiclient.jar file (for a WAS 4.0.x environment) is available in the appropriate directory.

- For UNIX, the file should be in the HOME/classes directory.
- For Windows, the file should be in the \candle\cma directory

KWE0024E **Unable to start profiler monitoring for server** ***servername*** **RC =** *return_code*.

Explanation: The OMEGAMON XE for WAS monitoring agent attempted to initiate profiler data collection, but a problems was detected. The return_Code indicates the type of error, which can be one of the following.

10	Network_Subsystem_Down
11	Socket_Not_Connected
12	Socket_Shutdown
13	Virtual_Circuit_Terminated
14	Virtual_Circuit_Reset
15	Unknown_Socket_Error
17	Socket_Timeout
18	Bas response received from the profiler.
19	Invalid_Data_Type_Received from the profiler
22	Permission_To_Create_Socket_Denied
23	Descriptor_Table_Full
24	Insufficient_Memory_For_Socket
25	Insufficient_Streams_Available
26	Unsupported_Protocol
27	Interrupted_By_Signal
28	Message_Too_Long
29	Socket_IO_Error

User Response: Attempt to determine why the error condition specified in the message occurred. Note: this error can occur if the application server was stopped. If a return code of 19 appears in the message, there should be another message in the log that further identifies the problem.

KWE0025E **Default WAS path is not set - EPM data not available.**

Explanation: The WASAppServerRoot parameter was not provided in the kwe.xml file. This parameter, which identifies the root directory where the WebSphere Application Server was installed, is required on UNIX systems.

User Response: Provide the WASAppServerRoot parameter in the kwe.xml file.

KWE0026E **Unable to find WAS path in the Windows registry.**

Explanation: The WASAppServerRoot parameter was not provided in the kwe.xml file, and the OMEGAMON XE for WAS monitoring agent was unable to locate the HKWY_LOCAL_MACHINE/SOFTWARE/IBM/WebSphere Application Server key in the Windows registry.

User Response: Either provide the value for the WebSphere Application Server install directory in the WASAppServerRoot parameter in the kwe.xml file, or ensure that the Windows registry correctly reflects the fact that WebSphere Application Server is installed.

KWE0027E **Error encountered processing configuration file - RC = *return_code*.**

Explanation: An error was encountered in processing the kwe.xml parameter file. The *return_code* variable indicates the type of error. The possible return codes are:

0	WE_Command_Successful
1	Rogue_Wave_Storage_Failure
2	Program_Exception_Shutdown
3	No_Storage_Available
4	Too_Many_Locks_Acquired
5	Exception_Caught
6	Deadlock_Condition_Averted
7	Cant_Find_Application_Server
8	Nothing_From_TakeSample
9	Cant_Find_Resource
10	Network_Subsystem_Down
11	Socket_Not_Connected
12	Socket_Shutdown
13	Virtual_Circuit_Terminated
14	Virtual_Circuit_Reset
15	Unknown_Socket_Error
16	Socket_Connection_Terminated

17	Socket_Timeout
18	Socket_Bad_Response
19	Invalid_DataType_Received
20	Profiling_Already_Active
21	Requested_Profiling_Not_Active
22	Permission_To_Create_Socket_Denied
23	Descriptor_Table_Full
24	Insufficient_Memory_For_Socket
25	Insufficient_Streams_Available
26	Unsupported_Protocol
27	Interrupted_By_Signal
28	Message_Too_Long
29	Socket_IO_Error
30	Socket_Request_Timeout
31	Java_Exception_Occurred
32	PushLocalFrame_Error
33	Socket_Is_Closed
34	EPMClient_Not_Available
35	AppServer_Excluded_From_Monitoring
36	Register_Subnode_Failure
37	Invalid_JVMSTAT_Command
38	JVMStart_Bad_AppServer_Name
39	Start_AppServer_Bad_Name
40	Stop_AppServer_Bad_Name
41	AppServer_Not_Running
42	File_Not_Found
43	Unable_To_Start_Process
44	Pipe_Failed

45	KRA_Framework_Error
46	Config_Parms_Missing
47	Not_Authorized
48	Cant_Find_WAS_Path
49	Cant_Locate_CreateJavaVM_Function
50	Cant_Find_Java_Shared_Library
51	Cant_Locate_EPMClient_Class
52	Invalid_Command
53	AppServer_Not_Stopped
54	Profiler_Not_Supported_On_This_Platform
55	Profiler_Library_Not_Installed
56	Conflicting_AppServer_Initialization_Parms
57	AppServer_Initialization_Parms_Already_Setup
58	Invalid_Profiler_Collection_Type
59	Cant_Find_String_Class
60	Cant_Build_String_Array
61	Recoverable_Java_Exception_Occurred
62	Incorrect_Number_Of_Parms
63	Invalid_Instrumentation_Level
64	Invalid_Instrumentation_Level_Type
65	Profiler_Stopped_DueTo_Class_Load
66	Profiler_Cant_Locate_Background_Environment
67	Profiler_Cant_Find_System_Class
68	Profiler_Cant_Find_GetProperties_Method
69	Profiler_Cant_Get_Compiler_Property
70	Profiler_Compiler_Property_Not_None
71	Kill_Pending_For_Another_Thread
72	Thread_Not_Found

73	Java_Exception_Occurred_In_App_Server
74	Profiler_Cant_Find_Thread_Class
75	Profiler_Cant_Find_Thread_Stop_Method
76	ThreadId_Missing
77	Cant_Copy_Properties_File
78	Cant_Open_Input_Properties_File
79	Cant_Open_Output_Properties_File
80	Unexpected_WSCP_Output
81	End_Of_File
82	Profiler_Cant_Run_On_This_Java
83	Unable_To_Start_Logstream
84	Unable_To_Get_Logstream_Data
85	Unable_To_Start_Logstream
86	Unable_To_Get_Logstream_Data
87	Unable_To_Start_Master_Console
88	Unable_To_Start_SMF_Collection
89	Cant_Find_Globals_Class
90	Cant_Find_Instrumenter_Class
91	Cant_Find_Logger_Class
92	Cant_Register_Native_Functions
93	Cant_Find_MaskList_Class
94	Cant_Find_Chooser_Class
95	Workload_Analysis_Not_Active
96	AppTrace_Not_Active
97	Invalid_CollectWorkload_Command
98	Invalid_StartAppTrace_Command
99	Invalid_StopAppTrace_Command
100	Invalid_DeleteAppTrace_Command

101	Invalid_SetExceptionWorkload_Command
102	Failed_To_Get_Trace_ID
103	Appplication_Trace_Already_Active
104	Profiler_Build_Level_Incompatible_With_Agent
105	Workload_Analysis_Build_Level_Incompatible_With_Agent
106	IOException_Creating_Application_Trace_File

User Response: Examine the return code in the message and take the appropriate action. If necessary, contact Candle Customer Support.

KWE0028W **Error *aaaa* getting samples for originnode *dddd* SampleType=*bbbb* Situation=*cccc* FilterInfo = *eeee ffff gggg hhhh*.**

Explanation: A problem was encountered returning data to the Candle Management Server. The variables in the message have the following meanings.

aaaa	Indicates the return code.
dddd	Identifies the application server.
bbbb	Indicates the type of data requested.
cccc	Indicates the name of the situation for which the data is requested.
eeee ***ffff*** ***gggg*** ***hhhh***	Indicate the additional filter parameters that were specified for the request.

User Response: Usually, this message is produced when a request is made for data that the agent is not currently collecting, or when a request is specified for a resource that does not exist.

KWE0029I **Profiler data collection modified for application server *servername*.**

Explanation: The OMEGAMON XE for WAS monitoring agent has successfully modified the profiler data collection parameters as a result of a JVMSTAT take action command.

User Response: None. This an informational message only.

KWE0030E **Function *function_name* failed - *return_code*.**

Explanation: The function that is indicated by the *function_name* variable encountered an error indicated by the *return_code* value.

User Response: This message reports the return code from the operating system. If you require more help, contact Candle Customer Support.

KWE0031E **Invalid Profiler data collection specification:** ***parameter*** **ignored.**
Explanation: You specified invalid profiler data collection parameters, either in a JVMSTAT take action command, or in the JVMProfilerCollect parameter in the kwe.xml configuration file. The *parameter* variable indicates the invalid parameter.
User Response: Correct the data collection parameters.

KWE0032W **Invalid configuration parameter** ***parameter_name*****.**
Explanation: An invalid configuration parameter indicated by the *value* variable was specified in the kwe.xml file. The *parameter_name* variable indicates the invalid parameter.
User Response: Correct the configuration parameter.

KWE0033W **Application Server** ***servername*** **referenced in configuration file but does not exist in WebSphere.**
Explanation: The kwe.xml configuration file contained a reference to an application server identified by the *servername* variable, but this application server does not exist in the WebSphere environment
User Response: This message is a warning. The configuration parameters pertaining to servername are ignored. If necessary, correct the kwe.xml file.

KWE0034E **Invalid StatisticInterval configuration value** ***nnn*** **for application server** ***servername*****. Default value** ***ddd*** **used.**
Explanation: An invalid value was specified for the StatisticInterval parameter in the kwe.xml configuration file for the application server, *servername*. The *nnn* variable indicates the invalid value.
User Response: Correct this parameter value in the kwe.xml file.

KWE0035E **Invalid JVMProfilerCollectStopAfter configuration value** ***nnn*** **for application server** ***servername*****. Default value** ***ddd*** **used.**
Explanation: An invalid value was specified for the JVMProfilerCollectStopAfter parameter in the kwe.xml configuration file for the application server, *servername*. The *nnn* variable indicates the invalid value. The indicated default value of this parameter is being used.
User Response: Correct this parameter value in the kwe.xml file.

KWE0036E **Invalid JVMProfilerSampleInterval configuration value *nnn* for application server *servername*. Default value *ddd* used.**

Explanation: An invalid value for the JVMProfilerSampleInterval parameter was specified in the kwe.xml configuration file for the application server identified by the *servername* variable. The *nnn* variable indicates the invalid value. The indicated default value of this parameter is being used.

User Response: Correct this parameter value in the kwe.xml file.

KWE0037E **Invalid JVMProfilerCPUSampleInterval configuration value *nnn* for application server *servername*. Default value *ddd* used.**

Explanation: An invalid value was specified for the JVMProfilerCPUSampleInterval parameter in the kwe.xml configuration file for the application server identified by the *servername* variable. The indicated default value of this parameter is being used.

User Response: Correct the value of the JVMProfilerCPUSampleInterval parameter in the kwe.xml file.

KWE0038E **Invalid LogScanInterval configuration value *nnn* for application server *servername*. Default value *ddd* used.**

Explanation: An invalid value was specified for the LogScanInterval parameter in the kwe.xml configuration file. The indicated default value of this parameter is being used.

User Response: Correct the value of the LogScanInterval parameter in the kwe.xml file.

KWE0039E **Invalid RetainLogFileEvents configuration value *nnn* for application server *servername*. Default value *ddd* used.**

Explanation: An invalid value was specified for the RetainLogEvents_ parameter in the kwe.xml configuration file for the application server, *servername*. The indicated default value of this parameter is being used.

User Response: Correct the value of this parameter in the kwe.xml file.

KWE0040E **Profiler data collection on application server *servername* ended because of load of class *class_name*.**

Explanation: The profiler running on application server, *servername*, stopped collecting data because it detected a Java class load for the *class-name* indicated in the message. On AIX, the profiler may stop collecting data as a result of a class load.

User Response: Use a JVMSTAT take action command to restart data collection by the profiler on this application server.

KWE0041E **Profiler data collection on application server *servername* can't be started - unable to locate background thread.**

Explanation: You attempted to kill a Java thread; however, the profiler could not find the requested thread.

User Response: None. This an informational message only.

KWE0042E **Profiler data collection on application server *servername* can't be started - unable to locate System class.**

Explanation: You attempted to start profiler data collection; however, the profiler was unable to locate the java.lang.System class.

User Response: Contact Candle Customer Support.

KWE0043E **Profiler data collection on application server *servername* can't be started - unable to locate System.getProperty method.**

Explanation: You attempted to start profiler data collection; however, the profiler was unable to locate the getProperty method of the java.lang.System class.

User Response: Contact Candle Customer Support.

KWE0044E **Profiler data collection on application server *servername* can't be started - unable to locate java.compiler property**

Explanation: You attempted to start profiler data collection; however, the profiler was unable to locate the java.compiler property to determine whether or not the JIT compiler is disabled.

User Response: Ensure that **-Djava.compiler=NONE** is specified on the application server command line arguments in order to run profiler data collection on Windows or AIX.

KWE0045E **Profiler data collection on application server *servername* can't be started - java.compiler property is not set to NONE**

Explanation: You attempted to start profiler data collection; however, the profiler determined that the java.compiler property is not set to a value of **NONE**.

User Response: Ensure that the -Djava.compiler=NONE is specified on the application server command line arguments in order to run profiler data collection on Windows or AIX.

KWE0046E **Thread on application server *servername* can't be killed - kill is pending for another thread**

Explanation: You attempted to kill a Java thread; however, the profiler cannot process the request because another kill request is pending for another thread.

User Response: None. This an informational message only.

KWE0047E **Thread on application server *servername* can't be killed - thread id not found**

Explanation: You attempted to kill a Java thread; however, the profiler could not find the requested thread.
User Response: None. This an informational message only.

KWE0049E **Thread on application server *servername* can't be killed - Thread class not found**

Explanation: You attempted to kill a Java thread; however, the profiler could not locate the java.lang.Thread class.
User Response: Contact Candle Customer Support.

KWE0050E **Thread on application server *servername* can't be killed - Thread.stop() method not found**

Explanation: You attempted to kill a Java thread; however, the profiler could not locate the **stop()** method for the java.lang.Thread class.
User Response: Contact Candle Customer Support.

KWE0051E **Unable to open WAS properties file *filename*. WAS version not determined**

Explanation: The OMEGAMON XE for WAS monitoring agent was unable to open the WebSphere product.xml file indicated. The agent is unable to determine the current version of WebSphere.
User Response: Ensure that the product.xml file exists in the WebSphere properties/com/iibm/websphere directory and that the agent has access to the file.

KWE0052E **Unable to open Java log file *filename*. No logging from EPM client will be done**

Explanation: The OMEGAMON XE for WAS monitoring agent was unable to open the log file identified by *filename* variable. This is a log file used by the OMEGAMON for WAS EPM/PMI data collection. Logging from this component will not be done.
User Response: Attempt to determine why the file identified in the message could not be opened.

KWE0055W ***cccc* exception caught in *unit-name function-name*.**

Explanation: This message indicates that a WARNING exception of the type indicated by the *cccc* variable has occurred in the OMEGAMON XE for WAS agent. The *unit-name* and *function-name* variables in the message

contain the names of the unit and the function in which the exception occurred.
User Response: This is a warning condition. If you cannot explain this error, contact Candle Customer Support.

KWE0056I *cccc* **exception caught in** *unit_name function_name*.
Explanation: This message indicates that an informational exception of the type indicated by the *cccc* variable has occurred in the OMEGAMON XE for WAS monitoring agent. The *unit_name* and *function_name* variables in the message contain the names of the unit and the function in which the exception occurred.
User Response: None. This an informational message only.

KWE0057I **Version** *vvv* **of WebSphere Application Server detected**
Explanation: The OMEGAMON XE for WAS agent has found and reported the version level of WebSphere Application Server currently installed.
User Response: None. This an informational message only.

KWE0058E **Can't create Java Virtual Machine. Error create temporary properties file:** *filename*
Explanation: The OMEGAMON XE for WAS monitoring agent was unable to create a temporary properties files identified by the *filename* variable. As a result, no WebSphere monitoring is possible.
User Response: Attempt to determine why the file identified by the error message cannot be created. This could be a problem related to authorization problem or a shortage of disk space.

KWE0060E **Console messages are not monitored - RC =** *return_code*
Explanation: The OMEGAMON XE for WAS monitoring agent was unable to initiate the monitoring of messages to the OS/390 system console. The return code value of 8 indicates that the attach failed.
User Response: Contact Candle Customer Support.

KWE0061E **Logstream collection error for logstream** *xxxxxxxx* **- RC =** *return_code*
Explanation: The OMEGAMON XE for WAS monitoring agent was unable to initiate the logstream browser for the logstream identified by the *xxxxxxxx* variable. The possible return codes are

12	Parameter error
16	Abend occurred

User Response: Contact Candle Customer Support.

KWE0062E **SMF collection cannot be started - RC =** *return_code*

Explanation: The OMEGAMON XE for WAS monitoring agent was unable to initiate System Management Facility (SMF) data collection for WebSphere. The possible return codes are

12	SMF data is not being collected.
16	Unsupported release of operating system
20	KWWASMFI failed
24	Unable to acquire storage
28	KWWASMFR failed
32	Failure in IEANTRT

User Response: Confirm that the following exit points are active in SMF: IEFU83, IEFU84, and IEFU85. If the problem continues, contact Candle Customer Support.

KWE0063E **SMF record buffer exceeded maximum limit of** *xxx* **records. Some data may be missing.**

Explanation: The OMEGAMON XE for WAS monitoring agent was unable to process all the System Management Facility (SMF) data written by WebSphere because of the limitation set for the maximum number of unprocessed records. This condition can be caused by a problem in the OMEGAMON XE for WAS agent that prevents the SMF data from being processed quickly enough.

User Response: Increase the maximum number of unprocessed SMF records by modifying the SMFMaxRecords attribute in the KWWXML member of the RKANDATV dataset. If the problem continues, contact Candle Customer Support.

KWE0064E **Default workload collection request failed with RC** *return_code*

Explanation: The workload request specified in the agent configuration file, KWWXML, failed with a return code specified by the return_code variable.

User Response: See the accompanying messages, KWE00783E or KWE74E for more details.

KWE0065I **Started collecting workload data for ClassType=***aaa* **ClassName=***bbb*

Explanation: This message indicates that the OMEGAMON XE monitoring agent successfully started collecting workload data for the ClassType and ClassName specified by the *aaa* and *bbb* variables.

User Response: None. This an informational message only.

KWE0066I **Started collecting exception workload data for WorkloadName=*aaa***

Explanation: This message indicates that the OMEGAMON XE monitoring agent successfully started collecting exception workload data for the WorkloadName specified by the *aaa* variable.

User Response: None. This an informational message only.

KWE0067I **Started application trace '*trace_name*'**

Explanation: This message indicates that in response to a take action command the OMEGAMON XE monitoring agent successfully started recording to the application trace file specified by the *trace_name* variable.

User Response: None. This an informational message only.

KWE0068E **Failed to start application trace '*trace_name*' with RC *return_code***

Explanation: This message indicates that in response to a take action command the OMEGAMON XE for WAS monitoring agent failed to start recording to the application trace file specified by the *trace_name* variable because of the code reported by the *return_code* variable.

User Response: Message KWE0024E lists the possible return codes. In addition, the following return codes are also possible.

103	AppTraceAlreadyActive
106	AppTrace_IOException
107	AppTrace_Failed_Getting_Summary

Choose one of these responses.

- If RC=103 is received, please retry at a later time; only one active trace is allowed.
- If RC=106 or RC=107 is received, a possible I/O error or file access problem is indicated.
- If any other return code is received, see the User Response section for KWE0024E.

KWE0069I **Application trace '*trace_name*' is deleted**

Explanation: This message indicates that in response to a take action command the OMEGAMON XE monitoring agent successfully deleted the application trace specified by the *trace_name* variable.

User Response: None. This an informational message only.

KWE0070E **Failed to delete application trace '*trace_name*' RC *return_code***

Explanation: This message indicates that in response to a take action command the OMEGAMON XE for WAS monitoring agent failed to delete the application trace file specified by the *trace_name* variable due to the code displayed by the *return_code* variable. The possible return code is

78	Can't_Open_Input_Prop_File

User Response: Verify that the file with the name specified by the trace_name variable exists and the agent has authority to delete it.

KWE0071I **Take Action command '*command_name*' completed successfully.**

Explanation: This message indicates that the take action command was successfully processed by the OMEGAMON XE for WAS monitoring agent.
User Response: None. This an informational message only.

KWE0072E **Take Action command '*command_name*' failed - RC *return_code***

Explanation: This message indicates that the take action command identified by the *command name* variable was not successfully processed by the OMEGAMON XE for WAS monitoring agent due to the code displayed by the *return_code* variable. If the RC is 21, the agent did not have a a session with the profiler when a take action command was entered for the profiler.
User Response: Analyze the preceding error messages and respond accordingly.

KWE0073E **Failed to start collecting workload data for ClassType=*aaa* ClassName=*bbb* with RC *return_code***

Explanation: This message indicates that the OMEGAMON XE monitoring agent failed to start collecting workload data for the ClassType and ClassName specified by the *aaa* and *bbb* variables due to the return code displayed by the *return_code* variable.
User Response: See the explanation of message KWE0024E for a list of possible return codes.

KWE0074E **Failed to start collecting workload data for WorkloadName=*aaa* with RC *return_code***

Explanation: This message indicates that the OMEGAMON XE monitoring agent failed to start collecting workload data for the WorkloadName specified by the *aaa* variable due to the code displayed by the *return_code* variable.

User Response: See the explanation of message KWE0024E for a list of possible return codes.

KWE0075E **Application tracing stop failed with RC** *return_code*

Explanation: This message indicates that in response to a take action command the OMEGAMON XE monitoring agent failed to stop application tracing due to the code displayed by the *return_code* variable.

User Response: See the explanation of message KWE0024E for a list of possible return codes.

KWE0076E **Deleting the application trace file** *file_name* **is not allowed. Trace file is for server instance** *instance1_name* **but request was issued against server instance** *instance2_name*

Explanation: The attempt to delete the application trace file indicated in the *file_name* variable was rejected because the file was produced for the server instance indicated by the *instance1_name* variable, but the request was issued against the server instance indicated in the *instance2_name* variable.

User Response: Retry the take action and specify the correct destination system.

KWE0077E **Unable to retrieve WebSphere configuration data from the System Management server. See SYSPRINT for more messages.**

Explanation: A problem has occurred attempting to start a connection with WebSphere's System Management server. A configuration or authorization problem is probably the likely cause.

User Response: Look for messages in SYSPRINT and attempt to determine the cause of the problem. Contact Candle Customer Support if you cannot resolve the problem.

KWE0078E **Unable to retrieve WebSphere configuration data.**

Explanation: This could be a communication or authorization problem with the System Management server.

User Response: See the subsequent messages for more details.

KWE0079E **Socket error detected. Function** *function_name* **RC** *return_code* **errno =** *error*

Explanation: The OMEGAMON XE agent encountered an error on the socket connection to a WebSphere Application Server JVN profiler or workload analysis component. The function_name indicates the type of socket operation and may be one of the following values:

- Send

- Receive
- Connect
- Listen
- Bind

The *return_code* indicates the type of error, and can be one of the following values:

10	Network subsystem down
11	Socket not connected
12	Socket shutdown
13	Virtual circuit terminated
14	Virtual circuit reset
15	Unknown socket error
16	Socket connection terminated
17	Socket timeout
18	Bad response from socket
19	Invalid data type received
22	Permission to create socket denied
23	Descriptor table full
24	Insufficient memory for socket
25	Insufficient streams available
26	Unsupported protocol
27	Interrupted by signal
28	Message too long
29	Socket I/O error
30	Socket request timeout
33	Socket is closed
37	Given address is already in use
108	Socket connection reset by peer

The *error* variable is the actual error code from the socket request. The OMEGAMON XE agent responds by terminating communication over the socket connection

User Response: Based upon the *return_code* value, attempt to determine what caused the problem. For example, an *return_code* value of 108 can be caused by the application server region being brought down. If necessary, contact Candle Customer Support.

KWEWA messages

The KWEWA messages are issued within the WebSphere Application Server address space. The following messages cover a numbering range of KWEWA001–KWEWA046E and are listed alphabetically.

KWEWA001E **AgentConnection class initializer not able to load native library:** *library_name.*

Explanation: The Candle Workload Analysis component was unable to load the native library *library_name*. The Workload Analysis component cannot continue.

User Response: Confirm that the library specified by library_name is present in the /CANDLEHOME/kww directory. If it is, check the environmental variables associated with the application server instance to ensure that the JVM_BOOTLIBRARYPATH is set up properly.

KWEWA002I **Candle Workload Analysis Enabler is being initialized.**

Explanation: The Candle Workload Analysis Enabler is starting it's initialization.

User Response: None. This an informational message only. More messages should follow.

KWEWA003I **Workload Analysis library(wa-jarfile-name) build level level-1 incompatible with Candle_WA_Enabler(scl-jarfile-name) build level level-2.**

Explanation: During initialization, the Workload Analysis Enabler found that the build level of it's jar file (level-2) is incompatible with the build level of the Workload analysis component jar file (level-1). The Workload Analysis component cannot continue.

User Response: If Candle maintenance was recently applied, ensure that all steps in the configuration process were completed successfully. Ensure also that you successfully ran the setupCandleWAEnabler.sh shell script from USS.

KWEWA004E -JavaRoot option is set twice.

Explanation: The -JavaRoot parameter was specified multiple times to the CandleWASetup utility.

User Response: Correct the error and rerun.

KWEWA005E -JavaRoot directory name is missing.

Explanation: The -JavaRoot parameter was specified with no value to the CandleWASetup utility.

User Response: Correct the error and rerun.

KWEWA006E -JarFile option is set twice.

Explanation: The -JarFile parameter was specified multiple times to the CandleWASetup utility.

User Response: Correct the error and rerun.

KWEWA007E -JarFile directory name is missing.

Explanation: The -JarFile parameter was specified with no value to the CandleWASetup utility.

User Response: Correct the error and rerun.

KWEWA008E -ClassLoader option is set twice.

Explanation: The -ClassLoader parameter was specified multiple times to the CandleWASetup utility.

User Response: Correct the error and rerun.

KWEWA009E -ClassLoader directory name is missing.

Explanation: The -ClassLoader parameter was specified with no value to the CandleWASetup utility.

User Response: Correct the error and rerun.

KWEWA010E -OutFile option is set twice.

Explanation: The -OutFile parameter was specified multiple times to the CandleWASetup utility.

User Response: Correct the error and rerun.

KWEWA011E -OutFile directory name is missing.

Explanation: The -OutFile parameter was specified with no value to the CandleWASetup utility.

User Response: Correct the error and rerun.

KWEWA012E -Replace option is set twice.

Explanation: The -Replace parameter was specified multiple times to the CandleWASetup utility.

User Response: Correct the error and rerun.

KWEWA013E **-Replace directory name is missing.**

Explanation: The -Replace parameter was specified with no value to the CandleWASetup utility.

User Response: Correct the error and rerun.

KWEWA014E **Unrecognized option:** ***option*****.**

Explanation: An unrecognized option was specified to the CandleWASetup utility.

User Response: Correct the error and rerun.

KWEWA015E **Required parameter missing: -JavaRoot.**

Explanation: The -JavaRoot parameter was not specified to the CandleWASetup utility. This parameter is required.

User Response: Correct the error and rerun.

KWEWA016E **Required parameter missing: -OutFile.**

Explanation: The -OutFile parameter was not specified to the CandleWASetup utility. This parameter is required.

User Response: Correct the error and rerun.

KWEWA017E **Required parameter missing: -Replace.**

Explanation: The -Replace parameter was not specified to the CandleWASetup utility. This parameter is required.

User Response: Correct the error and rerun.

KWEWA018E **Unable to instrument jar file filename because of ZipException exception.**

Explanation: The CandleWASetup utility failed because of a problem reading or writing a jar file. The specific reason for the failure is given in exception.

User Response: Correct the error and rerun.

KWEWA019E *exception*

Explanation: The CandleWASetup utility failed because of a Java exception. The specific reason for the failure is given by *exception* variable.

User Response: Correct the error and rerun. If necessary, call Candle customer support.

KWEWA020I **Candle workload analysis class loader hook is active.**

Explanation: The Candle workload analysis component class loader has been activated.

User Response: None. This an informational message only.

KWEWA022I Candle WAS Workload Analysis component has been initialized. Build Level: ***build-level***

Explanation: The Candle workload analysis component has been fully initialized. The build level of the component is given by *build-level* variable.

User Response: None. This an informational message only.

KWEWA023I Candle workload analysis collector is being activated.

Explanation: The Candle workload analysis component is being activated.

User Response: None. This an informational message only. The KWEWA022I message should follow shortly.

KWEWA024I The Candle JVM Profiler is present.

Explanation: The Candle workload analysis component has recognized that the Candle JVM profiler is running in the same Java Virtual Machine.

User Response: None. This an informational message only.

KWEWA025E Exception caught by instrumentClass.

Explanation: A Java exception was caught while attempting to perform instrumentation on a Java class. Details of the exception will appear in the Workload Analysis component log file.

User Response: If necessary, call Candle customer support.

KWEWA026I Candle workload analysis detailed logging is enabled.

Explanation: Based upon the contents of the XML configuration file(s), the Workload Analysis component has enabled detailed logging.

User Response: None. This an informational message only.

KWEWA027I Candle workload analysis detailed logging is disabled.

Explanation: Based upon the contents of the XML configuration file(s), the Workload Analysis component has disabled detailed logging.

User Response: None. This an informational message only.

KWEWA028E Unable to find com.candle.kwe.instrument.*control_file* - Candle workload analysis disabled.

Explanation: The Workload Analysis component was unable to locate the instrumentation control file. The Workload Analysis component cannot continue.

User Response: Check the environmental variables associated with the Application Server instance to ensure that the JVM_EXTRA_OPTIONS variable is set up properly.

KWEWA029W **Error parsing User control file *file_name*. User control file ignored.**

Explanation: The Workload Analysis component encountered an error parsing the user instrumentation control file identified by the *file_name* variable. The entire contents of the user instrumentation control file are ignored.

User Response: Check for the KWEWA030W message, which will indicate the exact cause of the error. Correct the user instrumentation control file.

KWEWA030W *parse-error.*

Explanation: This message indicates that a error took place parsing a user instrumentation control file. The exact error is given by the *parse-error* variable.

User Response: Correct the user instrumentation control file.

KWEWA031W **User configuration file *file_name* does not exist - ignored.**

Explanation: The user instrumentation control file file-name, identified by the UserConfigFile attribute of the main instrumentation control file, cannot be located. The Workload Analysis component continues instrumentation using only the main instrumentation control file

User Response: If you want to provide user-specific instrumentation parameters, you should create the file specified by file-name.

KWEWA032E **Error parsing control file *file_name*.**

Explanation: The Workload Analysis component encountered an error parsing the main instrumentation control file identified by the *file_name* variable. The Workload Analysis component cannot continue.

User Response: Check for the KWEWA033W message, which will indicate the exact cause of the error. Correct the instrumentation control file.

KWEWA033W *parse-error.*

Explanation: This message indicates that a error took place parsing the main instrumentation control file. The exact error is given by the *parse-error* variable.

User Response: Correct the instrumentation control file.

KWEWA034I **Candle workload analysis control file is *file_name.***

Explanation: This message indicates the name of an instrumentation control file which is being used by the Workload Analysis component.

User Response: None. This an informational message only.

KWEWA035I **Candle workload analysis control file *file_name* has been parsed.**

Explanation: This message indicates that the instrumentation control file file-name has been successfully parsed.

User Response: None. This an informational message only.

KWEWA036W **Class Definition is being ignored because *error.***

Explanation: This message indicates a Class definition tag in an instrumentation control file is being ignored because of a syntax error. The specific error is given by the *error* variable.

User Response: Correct the instrumentation control file.

KWEWA037I **ClassTypeDef Definition is being ignored because *error.***

Explanation: This message indicates a ClassTypeDef definition tag in an instrumentation control file is being ignored because of a syntax error. The specific error is given by the *error* variable.

User Response: Correct the instrumentation control file.

KWEWA038E **Incompatibility detected between Java library(*wa_jarfile_name*) and Native Library(*native_library_name*). Java library build level is level-1 Native library build level is level-2.**

Explanation: During initialization, the Workload Analysis component found that the build level of it's jar file (level-1) is incompatible with the build level of the Workload Analysis component native library (level-2). The Workload Analysis component cannot continue.

User Response: If Candle maintenance was recently applied, ensure that all steps in the configuration process were completed successfully.

KWEWA039E **Incompatibility detected between Native library(*native_library_name*) and Java Library(*wa_jarfile_name*). Native library build level is level-1 Java library build level is level-2.**

Explanation: During initialization, the Workload Analysis component found that the build level of the Workload analysis native library (level-1) is incompatible with the build level of the workload analysis jar file (level-2). The Workload Analysis component cannot continue.

User Response: If Candle maintenance was recently applied, ensure that all steps in the configuration process were completed successfully.

KWEWA040E **Error exception caught attempting to verify Native library compatibility.**

Explanation: An exception occurred when the Workload Analysis component was attempting to verify compatibility with the native library. The exact error is given by exception.

User Response: If Candle maintenance was recently applied, ensure that all steps in the configuration process were completed successfully. Otherwise, contact Candle Customer Support.

KWEWA042E **Initialization failed due to failure parsing control file.**
Explanation: The Workload Analysis component was unable to complete initialization because of an error parsing the instrumentation control file. The Workload Analysis component cannot continue.
User Response: Find the prior KWEWA032E and KWEWA033W to determine the exact source of the error. Correct the instrumentation control file.

KWEWA043E **Initialization failed due to exception:** *exception.*
Explanation: The Workload Analysis component was unable to complete initialization because of a Java exception. The specific exception is given by the *exception* variable.
User Response: Contact Candle Customer Support.

KWEWA044W **Unable to open log-file-name due to exception:** *exception.*
Explanation: The Workload Analysis component was unable open the log file specified by log-file-name because of the exception given by exception. The Workload Analysis component proceeds without a log file
User Response: Correct the problem noted by the *exception* variable.

KWEWA045I **Candle workload enabler completed initialization.**
Explanation: The Workload Analysis Enabler has successfully completed initialization.
User Response: None. This an informational message only.

KWEWA046E **Exception: exception caught initializing Candle workload enabler.**
Explanation: The Workload Analysis Enabler was unable to complete initialization because of a Java exception given by exception.
User Response: Contact Candle Customer Support.

B

Guide to Candle Customer Support

Introduction

Candle Corporation is committed to producing top-quality software products and services. To assist you with making effective use of our products in your business environment, Candle is also committed to providing easy-to-use, responsive customer support.

Precision, speed, availability, predictability—these terms describe our products and Customer Support services.

Included in this Guide to Candle Customer Support is information about the following:

Base Maintenance Plan

Overview

Candle offers a comprehensive Base Maintenance Plan to ensure that you realize the greatest value possible from your Candle software investments. We have more than 200 technicians providing support worldwide, committed to being responsive and to providing expedient resolutions to support requests. Technicians are available worldwide at all times during the local business day. In the event of an after-hours or weekend emergency, our computerized call management and forwarding system will ensure that a technician responds to Severity One situations within one hour. For customers outside of North America, after-hours and weekend support is provided in English language only by Candle Customer Support technicians located in the United States.

Telephone support

Candle provides consistently reliable levels of service—thanks to our worldwide support network of dedicated experts trained for specific products and operating systems. You will always work with a professional who truly understands your problem.

We use an online interactive problem management system to log and track all customer-reported support requests. We give your support request immediate attention by routing the issue to the appropriate technical resource, regardless of geographic location.

Level 0 Support is where your call to Candle Customer Support is first handled. Your support request is recorded in our problem management system, then transferred to the appropriate Level 1 support team. We provide Level 0 manual interaction with our customers because we support more than 170 products. We feel our customers would prefer personal interaction to a complex VRU or IVR selection menu.

Level 1 Support is the service provided for initial support requests. Our Level 1 team offers problem determination assistance, problem analysis, problem resolutions, installation assistance, and preventative and corrective service information. They also provide product usage assistance.

Level 2 Support is engaged if Level 1 cannot provide a resolution to your problem. Our Level 2 technicians are equipped to analyze and reproduce errors or to determine that an error is not reproducible. Problems that cannot be resolved by Level 2 are escalated to Candle's Level 3 R&D support team.

Level 3 Support is engaged if a problem is identified in Candle product code. At Level 3, efforts are made to provide error correction, circumvention or notification that a correction or circumvention is not available. Level 3 support provides available maintenance modifications and maintenance delivery to correct appropriate documentation or product code errors.

eSupport

In order to facilitate the support process, Candle also provides **eSupport**, an electronic full-service information and customer support facility, via the World Wide Web at www.candle.com/support/. **eSupport** allows you to open a new service request and update existing service requests, as well as update information in your customer profile. New and updated service requests are queued to a support technician for immediate action. And we can respond to your request electronically or by telephone—it is your choice.

eSupport also contains a continually expanding knowledge base that customers can tap into at any time for self-service access to product and maintenance information.

The Candle Web Site and **eSupport** can be accessed 24 hours a day, 7 days a week by using your authorized Candle user ID and password.

Description of Candle severity levels

Responses to customer-reported product issues and usage questions are prioritized within Candle according to Severity Code assignment. Customers set their own Severity Levels when contacting a support center. This ensures that we respond according to your individual business requirements.

Severity 1 Crisis A crisis affects your ability to conduct business, and no procedural workaround exists. The system or application may be down.

Severity 2 High A high-impact problem indicates significant business effect to you. The program is usable but severely limited.

Severity 3 Moderate A moderate-impact problem involves partial, non-critical functionality loss or a reasonable workaround to the problem. A "fix" may be provided in a future release.

Severity 4 Low A low-impact problem is a "how-to" or an advisory question.

Severity 5 Enhancement Request This is a request for software or documentation enhancement. Our business units review all requests for possible incorporation into a future release of the product.

Candle has established the following service-level objectives:

Call Status	Severity 1 Goal	Severity 2 Goal	Severity 3 Goal	Severity 4 Goal	Severity 5 Goal
First Call Time to Answer	90% within one minute				
Level 1 Response (Normal Business Hours)	90% within 5 minutes	90% within one hour			
Level 2 Response (Normal Business Hours)	Warm Transfer	90% within two hours	90% within eight hours		
Scheduled follow-up (status update)	Hourly or as agreed	Daily or as agreed	Weekly or as agreed		Notification is made when an enhancement is incorporated into a generally available product.
	Notification is made when a fix is incorporated into a generally available product.				

The above information is for guideline purposes only. Candle does not guarantee or warrant the above service levels. This information is valid as of October 1999 and is subject to change without prior notice.

Recording and Monitoring Calls for Quality Purposes

Candle is committed to customer satisfaction. To ensure that our customers receive high levels of service, quality and professionalism, we'll monitor and possibly record incoming and outgoing Customer Support calls. The information gleaned from these calls will help us serve you better. If you prefer that your telephone call with Candle Customer Support in North America not be monitored or recorded, please advise the representative when you call us at **(800) 328-1811** or **(310) 535-3636**.

Customer Support Escalations

Candle Customer Support is committed to achieving high satisfaction ratings from our customers. However, we realize that you may occasionally have support issues that need to be escalated to Candle management. In those instances, we offer the following simple escalation procedure:

If you experience dissatisfaction with Candle Customer Support at any time, please escalate your concern by calling the Candle support location closest to you. Ask to speak to a Customer Support manager. During standard business hours, a Customer Support manager will be available to talk with you or will return your call. If you elect to hold for a manager, you will be connected with someone as soon as possible. If you wish a return call, please tell the Candle representative coordinating your call when you will be available. After contacting you, the Customer Support manager will develop an action plan to resolve your issue. All escalations or complaints received about support issues are logged and tracked to ensure responsiveness and closure.

Above and Beyond

What differentiates Candle's support services from our competitors? We go the extra mile by offering the following as part of our Base Maintenance Plan:

- Unlimited multi-language defect, installation and operations support
- eSupport using the World Wide Web
- Regularly scheduled product updates and maintenance provided at no additional charge
- Over 200 specialized technicians providing expert support for your Candle products

Enhanced Support Services

Overview

Our Base Maintenance Plan provides a high level of software support in a packaged offering. However, in addition to this plan, we have additional fee-based support services to meet unique customer needs.

The following are some examples of our added-value support services:

- **Assigned Support Center Representative Services (ASCR)**
  - An assigned focal point for managing support escalation needs
  - Proactive notification of available software fixes
  - Proactive notification of product version updates
  - Weekly conference calls with your ASCR to review active problem records
  - Monthly performance reviews of Candle Customer Support service levels
  - Optional on-site visits (extra charges may apply)
- **Maintenance Assessment Service (MAS)**
  - On-site assessment services
  - Advice about product maintenance and implementation
  - Training your staff to develop efficient and focused procedures to reduce overall cost of ownership of your Candle software products
  - Analysis of your Candle product environment: versions, updates, code correction history, incident history and product configurations
  - Reviews to ensure that purchased Candle products and solutions are used effectively
- **Multi-Services Manager (MSM)**

  Multi-Services Manager provides highly valued services to customers requiring on-site full time expertise to complement their technical resources.

  - Dedicated on-site Candle resource (6 months or one year) at your site to help ensure maximum use and effectiveness of your Candle products

- – Liaison for all Candle product support activities, coordination and assistance with implementation of all product updates and maintenance releases
- – Works with your staff to understand business needs and systems requirements
- – Possesses technical and systems management skills to enhance your staff’s knowledge and expertise
- – Other projects as defined in Statement of Work for MSM services

Customer Support Contact Information

Link to Worldwide Support Telephone and E-mail information

To contact Customer Support, the current list of telephone numbers and e-mail addresses can be found on the Candle Web site, www.candle.com/support/.

Select **Support Contacts** from the list on the left of the page.

Glossary

C

class file A file containing Java object code for a single Java class.

class loader A Java component that is responsible for loading Java classes.

CORBA Common Object Request Broker Architecture. A specification that provides standards for different types of object request brokers. The specification enables object-request brokers from different software vendors to work together.

CPU Central Processing Unit. The part of the computer that controls the interpretation and execution of instructions.

CT Candle Technologies. This is the foundation that supports the operations of CandleNet Portal and its OMEGAMON XE agents.

D

datasources Data pertaining to J2EE data sources, which are logical connections to database subsystems.

delay A measurable factor that contributes to the response time of a workload, for example, reading a row from an SQL database. Workloads themselves can be delays to other workloads, for example, a servlet invoking an EJB method. (See also workload.)

DCE Distributed Computing Environment. A specification from the Open Software Foundation (OSF) that provides networking functions.

WebSphere Application Server can use DCE to provide certain types of security, for example, authentication or tokens, tickets, or credentials in an untrusted network.

D

EJB Enterprise Java Bean. A component-application programming interface that defines component architecture for multi-tier client-server systems.

H

heap A memory storage area known as the System Resources area that is used to store system information about running applications.

HTTP Hypertext Transfer Protocol. A suite of protocols for the Internet that transfer and display hypertext documents.

HTTP sessions Data related to sessions with specific browsers.

I

instrumentation Modifications made to program code so that the amount of time it takes the code to run, along with other statistics, can be measured.

J

J2EE Java 2 Platform, Enterprise Edition. An environment for developing and deploying enterprise applications. The J2EE platform consists of a set of services, application programming interfaces (APIs), and protocols that provide the functionality for developing multi-tiered, web-based applications.

JAR file A Java archive file, which is a ZIP file containing many class files along with the directory structure.

JVM Java Virtual Machine. A software execution engine that safely and compatibly executes the byte codes in Java class files on a microprocessor, whether in a computer or in another electronic device.

JMS Java Messaging Service. A programming interface that connects Java programs to middleware messaging applications.

JMS summary data Data related to the use of MQSeries by WebSphere applications.

JNDI Java Naming and Directory Interface. A programming interface that connects to naming and directory services such as LDAP.

JVM Java Virtual Machine. A software implementation of a central processing unit (CPU) that runs Java applets and applications.

JVPMI Java Virtual Machine Profiling Interface. An experimental interface that provides profiling hooks in the Java virtual machine. This interface provides two-way function calls between the Java virtual machine and a profiler agent. The virtual machine notifies a profiler agent of various events and the agent issues control requests.

K

Keberos A security system from the Massachusetts Institute of Technology that provides security services for networking.

L

LDAP Lightweight Directory Access Protocol. A protocol that uses TCP/IP to access directories. Applications can use these directories for storing common data and for retrieving data. For example, applications can use LDAP to access such information as e-mail address, service configuration parameters, or public keys.

logstream A system logger application that records error information, unexpected condition or failure that is detected within WebSphere Application Server for OS/390 code.

longest running workloads Data related to individual transactions (executions of a workload) that exceed user-defined response time criteria.

M

MOFW server Managed-object framework server. A type of server that supports CORBA applications for WebSphere Application Server for OS/390.

monitor An transaction environment for maintaining large quantities of data in a consistent state and controlling which users and clients access data through defined servers.

N

node Any managed system, such as a MOFA or JAVA server, that CandleNet Portal is monitoring. A node can also be a managed system of sub-nodes being managed under a primary node.

O

OLT Object Level Trace. A tool for testing distributed applications that includes a graphical trace facility and a remote debugger.

R

RMI Remote Method Invocations. A Java standard that performs a remote procedure call (RPC), which allows Java objects stored in the network to be run remotely.

S

servlet A Java application that runs in an application server or a Web-server. It provides server-side processing.

signature A name of an operation and its parameters.

SMF System Management Facility. A feature of OS/390 that collects and records system and job-related information.

SSL Secure Sockets Layer. A security protocol for communication privacy to provide secure client-server conversations.

T

thread A dispatchable unit of work

W

workload The type of work whose response time directly reflects the response time perceived by an end user. In OMEGAMON XE for WebSphere Application Server for OS/390, the workload is a servlet or EJB business method. (See also delay.)

Index

User Comment Form

OMEGAMON® XE for WebSphere Application Server for OS/390 User's Guide, Version 110

Document Number WW54-6554-3

Please take a moment to share your comments and suggestions regarding Candle's documentation. Be as specific as possible. Use additional paper if necessary.

	Page #	Comment
1	______	______________________________
2	______	______________________________
3	______	______________________________
4	______	______________________________

Additional general comments to help us improve the documentation.

____

____

____

Please provide the following information. Thank you for your time and assistance.

____

Name **Job Function**

____

Company **Site ID**

____

Address

____

City, State, Zip

You may return your comments to us at your convenience.

FAX

(310) 727-4131

Fold, tape, and mail
(postage paid)
Please do not staple.

May we call to discuss your comments?

□ YES Phone number: ________________ Best time to call:________

□ NO

Cut or fold
along line

Fold and tape Please do not staple Fold and tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO 167 EL SEGUNDO CA

POSTAGE WILL BE PAID BY ADDRESSEE

Candle

ATTN: INFORMATION DEVELOPMENT
CANDLE CORPORATION
201 NORTH DOUGLAS STREET
EL SEGUNDO CALIFORNIA 90245-9796

Fold and tape Please do not staple Fold and tape

Cut or fold
along line