IBM Tivoli OMEGAMON XE for WebSphere Business Integration

Version 1.1.0

Configuring IBM Tivoli OMEGAMON XE for WebSphere Business Integration on z/OS
Configuring IBM Tivoli OMEGAMON XE for WebSphere Business Integration on z/OS
Before using this information and the product it supports, read the information in “Notices” on page 233.
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preface</td>
<td>13</td>
</tr>
<tr>
<td>Preface</td>
<td>15</td>
</tr>
<tr>
<td>Documentation Conventions</td>
<td>19</td>
</tr>
<tr>
<td>Chapter 1. Starting Point: Checklists</td>
<td>21</td>
</tr>
<tr>
<td>Using the Checklists and Topics in This Chapter</td>
<td>22</td>
</tr>
<tr>
<td>Checklist: Prerequisites for Configuring the Component Product</td>
<td>23</td>
</tr>
<tr>
<td>Checklist: Setting up the Configuration Environment</td>
<td>24</td>
</tr>
<tr>
<td>Checklists: Setting up a Runtime Environment and Configuring the Component Product in It</td>
<td>25</td>
</tr>
<tr>
<td>Checklist: Configuring IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration in a Runtime Environment</td>
<td>31</td>
</tr>
<tr>
<td>Checklist: Configuring IBM Tivoli OMEGAMON XE for WebSphere MQ Monitoring in a Runtime Environment</td>
<td>35</td>
</tr>
<tr>
<td>Checklist: Configuring IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers in a Runtime Environment</td>
<td>38</td>
</tr>
<tr>
<td>Checklist: Configuring IBM Tivoli OMEGAMON XE for WebSphere MQ outside the Configuration Tool</td>
<td>41</td>
</tr>
<tr>
<td>Checklist: Configuring IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers outside the Configuration Tool</td>
<td>43</td>
</tr>
<tr>
<td>Checklist: Updating and Changing the Configuration</td>
<td>45</td>
</tr>
<tr>
<td>Checklist: Using Advanced Features Available in the Configuration Tool</td>
<td>46</td>
</tr>
<tr>
<td>Chapter 2. Overview of Processes, Methods, and the Configuration Tool Modes</td>
<td>47</td>
</tr>
<tr>
<td>Terminology</td>
<td>48</td>
</tr>
<tr>
<td>Configuration Tool Features</td>
<td>49</td>
</tr>
<tr>
<td>Configuration Tool Modes: Interactive and Batch</td>
<td>51</td>
</tr>
<tr>
<td>Chapter 3. Planning the Configuration</td>
<td>53</td>
</tr>
<tr>
<td>Introduction to OMEGAMON XE and OMEGAMON DE</td>
<td>54</td>
</tr>
<tr>
<td>Requirements to Review Before Configuring</td>
<td>60</td>
</tr>
<tr>
<td>Considerations and Requirements for IBM Tivoli OMEGAMON XE for WebSphere MQ</td>
<td>62</td>
</tr>
<tr>
<td>Planning for Setting up a Runtime Environment and Configuring the Component Product in It</td>
<td>64</td>
</tr>
<tr>
<td>Chapter 4. Setting up the Configuration Environment and Starting the Configuration Tool</td>
<td>75</td>
</tr>
<tr>
<td>Setting up the Configuration Environment and Starting the Configuration Tool</td>
<td>76</td>
</tr>
<tr>
<td>Chapter 5. Setting up the Runtime Environment</td>
<td>79</td>
</tr>
<tr>
<td>Overview of Runtime Environment Configuration Sequences</td>
<td>80</td>
</tr>
<tr>
<td>Chapter 9.</td>
<td>Updating and Changing the Configuration</td>
</tr>
<tr>
<td>Chapter 10.</td>
<td>Enabling System Variable Support</td>
</tr>
<tr>
<td>Chapter 11.</td>
<td>Batch Mode Processing</td>
</tr>
<tr>
<td>Appendix A.</td>
<td>Recommended Configuration Scenarios</td>
</tr>
<tr>
<td>Appendix B.</td>
<td>Configuration Services and Utilities</td>
</tr>
<tr>
<td>Appendix C.</td>
<td>Configuration Tool Batch Utilities</td>
</tr>
<tr>
<td>Table 1.</td>
<td>Symbols in Command Syntax</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Table 2.</td>
<td>Checklists and Topics in This Chapter</td>
</tr>
<tr>
<td>Table 3.</td>
<td>Prerequisites for Configuring the Component Product</td>
</tr>
<tr>
<td>Table 4.</td>
<td>Setting up the Configuration Environment</td>
</tr>
<tr>
<td>Table 5.</td>
<td>Setting up a Runtime Environment and Configuring the Component Product in It</td>
</tr>
<tr>
<td>Table 6.</td>
<td>Setting up a New Runtime Environment and Configuring the Component Product in It</td>
</tr>
<tr>
<td>Table 7.</td>
<td>Setting up an Existing Runtime Environment and Configuring the Component Product in It</td>
</tr>
<tr>
<td>Table 8.</td>
<td>Configuring IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration in a Runtime Environment</td>
</tr>
<tr>
<td>Table 9.</td>
<td>Configuring IBM Tivoli OMEGAMON XE for WebSphere MQ Monitoring in a Runtime Environment</td>
</tr>
<tr>
<td>Table 10.</td>
<td>Configuring IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers in a Runtime Environment</td>
</tr>
<tr>
<td>Table 11.</td>
<td>Configuring IBM Tivoli OMEGAMON XE for WebSphere MQ outside the Configuration Tool</td>
</tr>
<tr>
<td>Table 12.</td>
<td>Configuring IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers outside the Configuration Tool</td>
</tr>
<tr>
<td>Table 13.</td>
<td>Updating and Changing the Configuration</td>
</tr>
<tr>
<td>Table 14.</td>
<td>Using Advanced Features Available in the Configuration Tool</td>
</tr>
<tr>
<td>Table 15.</td>
<td>Terminology</td>
</tr>
<tr>
<td>Table 16.</td>
<td>Commands and Functions</td>
</tr>
<tr>
<td>Table 17.</td>
<td>Configuration Tool Modes</td>
</tr>
<tr>
<td>Table 18.</td>
<td>Details about the Candle Management Server Types and Their Relationships</td>
</tr>
<tr>
<td>Table 19.</td>
<td>Runtime Environment Terminology</td>
</tr>
<tr>
<td>Table 20.</td>
<td>Additional Features and Enhancements for Configuring Runtime Environments</td>
</tr>
<tr>
<td>Table 21.</td>
<td>Actions Available from the Runtime Environments Panel</td>
</tr>
<tr>
<td>Table 22.</td>
<td>FULL Runtime Environment Example</td>
</tr>
<tr>
<td>Table 23.</td>
<td>BASE Runtime Environment Example</td>
</tr>
<tr>
<td>Table 24.</td>
<td>SHARING BASE Runtime Environment Example</td>
</tr>
<tr>
<td>Table 25.</td>
<td>SHARING FULL Runtime Environment Example</td>
</tr>
<tr>
<td>Table 26.</td>
<td>SHARING SMP/E TARGET Runtime Environment Example</td>
</tr>
<tr>
<td>Table 27.</td>
<td>Set Up Configuration Environment Panel</td>
</tr>
<tr>
<td>Table 28.</td>
<td>Runtime Environment Types</td>
</tr>
<tr>
<td>Table 29.</td>
<td>First Add Runtime Environment Panel</td>
</tr>
<tr>
<td>Table 30.</td>
<td>Second Add Runtime Environment Panel</td>
</tr>
<tr>
<td>Table 31.</td>
<td>Specify Configuration Parameters Panel</td>
</tr>
<tr>
<td>Table 32.</td>
<td>Persistent Datastore Specifications Panel</td>
</tr>
<tr>
<td>Table 33.</td>
<td>Specify Persistent Datastore Values Panel</td>
</tr>
<tr>
<td>Table 34.</td>
<td>Specify Configuration Parameters Panel</td>
</tr>
</tbody>
</table>
Figures

Figure 1. OMEGAMON XE Platform Components ................................................. 55
Figure 2. Example Configuration Including a Remote CMS ................................. 59
Figure 3. FULL Runtime Environment ............................................................... 69
Figure 4. BASE Runtime Environment .............................................................. 70
Figure 5. Runtime Environment SHARING with BASE RTE2 .............................. 71
Figure 6. Runtime Environment SHARING with FULL Runtime Environment1 ........ 72
Figure 7. Runtime Environment SHARING with SMP/E Target Libraries ................. 73
Figure 8. Process for Verifying the Configuration: Same Address Space .................. 141
Figure 9. Process for Verifying the Configuration: Different Address Space .............. 142
Figure 10. Process for Verifying the Configuration: CMS on a Different Platform ........ 143
Figure 11. Process for Updating and Changing the Configuration ......................... 168
Figure 12. Two Systems with 6 LPARs in Scenario 1 ........................................ 197
Figure 13. Two Systems with 6 LPARs in Scenario 2 ........................................ 200
Figure 14. Two Systems with 6 LPARs in Scenario 3 ........................................ 202
Figure 15. Two Systems with 6 LPARs in Scenario 4 ........................................ 205
Figure 16. Two Systems with 6 LPARs in Scenario 5 ........................................ 207
Figure 17. Copying Configuration Values from One Configuration Tool to Another .... 209
Figure 18. Example of a Batch Parameter Help .................................................. 223
Introduction

The IBM® Tivoli® OMEGAMON® XE for WebSphere® Business Integration package includes these component products:

- **IBM Tivoli OMEGAMON XE for WebSphere MQ**
  
The IBM Tivoli OMEGAMON XE for WebSphere MQ component product consists of these component products:
  
  - **IBM Tivoli OMEGAMON XE for WebSphere MQ Monitoring** is an agent that lets you easily collect and analyze WebSphere MQ-specific data for all the remote and local queue managers from a single vantage point.
  
  - **IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration** is an agent that helps you simplify the time-consuming and resource-intensive tasks of defining and managing a WebSphere MQ configuration.

- **IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers**

  IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers is an agent and plug-in node that lets you easily collect and analyze data from any of the message broker products for the WebSphere Business Integration package in order to help with the development, testing, management and troubleshooting of message flows.

*Configuring IBM Tivoli OMEGAMON XE for WebSphere Business Integration on z/OS®* describes how to configure and customize IBM Tivoli OMEGAMON XE for WebSphere MQ and IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers on the z/OS operating system.

Preface contents

- About This Guide ................................................................. 15
- Who should read this guide .................................................. 15
- Document set information ................................................... 15
- Tivoli technical training ................................................. 18
- Support information .......................................................... 18
- Documentation Conventions .............................................. 19
  
  - Overview ................................................................. 19
  - Panels and figures ...................................................... 19
  - Required blanks ....................................................... 19
  - Revision bars ........................................................... 19
About This Guide

Who should read this guide

This guide is for IT operations staff or administrators who are responsible for the following tasks:

- Installation of applications
- Automation of tasks on the system
- Monitoring new applications
- Trouble-shooting and providing solutions for operators when they have problems
- Fine-tuning the performance of systems (by measuring system capabilities and tweaking configuration settings)

Readers should be familiar with the following topics:

- z/OS operating system
- IBM’s WebSphere MQ product
- IBM’s message broker product
- The planned configuration for their OMEGAMON Platform and CandleNet Portal® environment. They should consult with their IBM Tivoli system administrator to ensure that they know where the Candle Management Server® (CMS), CandleNet Portal Servers, Candle Management Workstation® (CMW), CandleNet Portal, and the agents are to be installed.

Note: Before you can configure IBM Tivoli OMEGAMON XE for WebSphere Business Integration, you must have the OMEGAMON Platform installed and configured in the enterprise. For instructions, see “Installing and Setting up OMEGAMON Platform and CandleNet Portal on Windows and UNIX,” which is on the “OMEGAMON Platform version 360 and CandleNet Portal Documentation CD” that accompanied this component product of the IBM Tivoli OMEGAMON XE for WebSphere Business Integration package. In addition, you need to have IBM Tivoli OMEGAMON XE for WebSphere Business Integration installed. For instructions, see the “Program Directory” for the IBM Tivoli OMEGAMON XE for WebSphere Business Integration package.

Document set information

This section lists publications in the IBM Tivoli OMEGAMON XE for WebSphere Business Integration version 1.1.0 Documentation CD and the OMEGAMON Platform version 360 and CandleNet Portal Documentation CD that supply the following information:

- Installation and configuration of the IBM Tivoli OMEGAMON XE for WebSphere MQ component products on other platforms
- Operation of IBM Tivoli OMEGAMON XE for WebSphere MQ component products
About This Guide

- Installation, configuration, and use of the prerequisite OMEGAMON Platform and CandleNet Portal component products

The documentation CDs contain the publications that are in the package’s library. The format of the publications is PDF. Refer to the readme file on the CDs for instructions on how to access the documentation.

This section also lists other useful related documents. It also describes how to access Tivoli publications online and how to order Tivoli publications.

**IBM Tivoli OMEGAMON XE for WebSphere Business Integration documentation CD**

The following are useful documents that are available in the *IBM Tivoli OMEGAMON XE for WebSphere Business Integration version 1.1.0 Documentation CD*:

- **Using IBM Tivoli OMEGAMON XE for WebSphere MQ Monitoring**, SC31-6888-00, provides information about using IBM Tivoli OMEGAMON XE for WebSphere Business Integration in the CandleNet Portal interface.
- **Using IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration**, SC31-6889-00, provides information about the processes involved in preparing IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration to help you design and prototype WebSphere MQ resources.
- **Using IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers**, SC31-6890-00, provides information about using IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers in the CandleNet Portal interface.
- **Installing and Setting up IBM Tivoli OMEGAMON XE for WebSphere Business Integration on Windows and UNIX**, SC31-6885-00, provides information about installing and setting up IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers, and IBM Tivoli OMEGAMON XE for WebSphere InterChange Server on the Windows and UNIX platforms.
- **Installing and Setting up IBM Tivoli OMEGAMON XE for WebSphere MQ on OS/400**, SC31-6886-00, provides information about installing and setting up IBM Tivoli OMEGAMON XE for WebSphere MQ on the OS/400 platform.
- **Installing and Setting up IBM Tivoli OMEGAMON XE for WebSphere MQ on HP NonStop Kernel**, SC31-6887-00, provides information about installing and setting up IBM Tivoli OMEGAMON XE for WebSphere MQ on the HP NonStop Kernel platform.

**OMEGAMON Platform and CandleNet Portal documentation CD**

The following are useful documents that are available in the *OMEGAMON Platform version 360 and CandleNet Portal Documentation CD*:

- **Administering OMEGAMON Products: CandleNet Portal**, GC32-9180, describes the support tasks and functions required for the OMEGAMON platform, including CandleNet Portal user administration.
- **Using OMEGAMON Products: CandleNet Portal**, GC32-9182, describes the features of CandleNet Portal and how best to use them with OMEGAMON products.
Historical Data Collection Guide for IBM Tivoli OMEGAMON XE Products, GC32-9429-00, describes the process of collecting historical data and either warehousing it or converting it to delimited flat files for reporting purposes.

OMEGAMON Platform Messages manuals provide lists of descriptions that help you to interpret messages that are issued by the component products of the OMEGAMON Platform: CMS, CandleNet Portal, CMW, Warehouse Proxy, Alert Adapter for AF/REMOTE, Alert Adapter for Tivoli Enterprise Console, and Alert Emitter for Tivoli Enterprise Console on Windows and UNIX.

The following are the volumes:

- **IBM Tivoli Candle Products Messages Volume 1 (AOP–ETX)**, SC32-9416-00
- **IBM Tivoli Candle Products Messages Volume 2 (EU–KLVGM)**, SC32-9417-00
- **IBM Tivoli Candle Products Messages Volume 3 (KLVHS-KONCT)**, SC32-9418-00
- **IBM Tivoli Candle Products Messages Volume 4 (KONCV-OC)**, SC32-9419-00
- **IBM Tivoli Candle Products Messages Volume 5 (ODC–VEB and Appendixes)**, SC32-9420-00

Installing and Setting up OMEGAMON Platform and CandleNet Portal on Windows and UNIX, SC32-1768-00, provides information about installing and setting up the following component products of the OMEGAMON Platform: CMS, CandleNet Portal, CMW, Warehouse Proxy, Alert Adapter for AF/REMOTE, Alert Adapter for Tivoli Enterprise Console, and Alert Emitter for Tivoli Enterprise Console on Windows and UNIX.

The online glossary for the CandleNet Portal includes definitions for many of the technical terms related to OMEGAMON XE software.

**Accessing publications online**

IBM posts publications for this and all other Tivoli products, as they become available and whenever they are updated, to the Tivoli software information center Web site. Access the Tivoli software information center by first going to the Tivoli software library at the following Web address:


Scroll down and click the **Product manuals** link. In the Tivoli Technical Product Documents Alphabetical Listing window, click the link for IBM Tivoli OMEGAMON XE for WebSphere Business Integration to access the package’s library at the Tivoli software information center.

If you print PDF documents on other than letter-sized paper, set the option in the **File -> Print** window that allows Adobe Reader to print letter-sized pages on your local paper.

**Ordering publications**

You can order many Tivoli publications online at the following Web site:

You can also order by telephone by calling one of the following numbers:

- In the United States: 800-879-2755
- In Canada: 800-426-4968

In other countries, see the following Web site for a list of telephone numbers:

http://www.ibm.com/software/tivoli/order-lit

**Tivoli technical training**

For Tivoli technical training information, refer to the following IBM Tivoli Education Web site:

http://www.ibm.com/software/tivoli/education

**Support information**

If you have a problem with the IBM software, you want to resolve it quickly. IBM provides the following ways for you to obtain the support you need:

- Searching knowledge bases: You can search across a large collection of known problems and workarounds, Technotes, and other information.
- Obtaining fixes: You can locate the latest fixes that are already available for a component product.
- Contacting IBM Software Support: If you still cannot solve the problem, and you need to work with someone from IBM, you can use a variety of ways to contact IBM Software Support.

For more information about these three ways of resolving problems, see “Support Information” on page 227.
Documentation Conventions

Overview
This guide uses several conventions for special terms and actions, and operating system-dependent commands and paths.

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

Required blanks
The slashed-b (§) character in examples represents a required blank. The following example illustrates the location of two required blanks:
§eBA*ServiceMonitor§0990221161551000

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

Variables and literals
In examples of z/OS command syntax, uppercase letters are actual values (literals) that the user should enter; 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 enter LOGON APPLID followed by an application identifier (represented by cccccc) within parentheses.

Symbols
The following symbols may appear in command syntax:

<table>
<thead>
<tr>
<th>Table 1. Symbols in Command Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symbol</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>In this example, YES or NO may be specified.</td>
</tr>
<tr>
<td>[ ]</td>
</tr>
<tr>
<td>In this example, DEST is a required argument and ALTDEST is optional.</td>
</tr>
</tbody>
</table>
### Table 1. Symbols in Command Syntax (continued)

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>{ }</code></td>
<td>Some documents use braces to denote required arguments, or to group arguments for clarity. Example: &lt;br&gt;`COMPARE {workload} - REPORT={SUMMARY</td>
</tr>
<tr>
<td><code>_</code></td>
<td>Default values are underscored. Example: &lt;br&gt;`COPY infile outfile - [COMPRESS={YES</td>
</tr>
</tbody>
</table>
Starting Point: Checklists

Introduction

This chapter contains a checklist or topic for each of the steps in the process for configuring products that you can use for the following tasks:

- Guide you through the steps in the process for configuring and customizing products and components
- Verify that you have used all of the options in the Configuration tool required for your specific situation
- Verify that you have completed all the required steps outside of the Configuration tool to configure and customize the products and components
- Locate detailed information when you need background information or step-by-step instructions for a specific task

Chapter contents

Using the Checklists and Topics in This Chapter ..................................................22
Checklist: Prerequisites for Configuring the Component Product .............................23
Checklist: Setting up the Configuration Environment .............................................24
Checklist: Setting up a Runtime Environment and Configuring the Component Product in It ..........................................................25
Checklist: Configuring IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration in a Runtime Environment .............................................................31
Checklist: Configuring IBM Tivoli OMEGAMON XE for WebSphere MQ Monitoring in a Runtime Environment .............................................................35
Checklist: Configuring IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers in a Runtime Environment .............................................................38
Checklist: Configuring IBM Tivoli OMEGAMON XE for WebSphere MQ outside the Configuration Tool .................................................................41
Checklist: Configuring IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers outside the Configuration Tool .................................................................43
Checklist: Updating and Changing the Configuration ..............................................45
Checklist: Using Advanced Features Available in the Configuration Tool .................46
Using the Checklists and Topics in This Chapter

The tasks in the checklists are listed in the order they are performed. Use the last column to locate step-by-step instructions and background information in the resources provided with the component products (such as the Program Directory, this guide, or another guide).

Determining the checklists and topics to use

This chapter contains a checklist for each of the steps in the process of maintaining and configuring component products. In many cases, a succeeding chapter will correspond directly to a particular checklist in this chapter.

The following table shows the checklist or topic you should use for the tasks:

Table 2. Checklists and Topics in This Chapter

<table>
<thead>
<tr>
<th>Task</th>
<th>Checklists and Topics to Use in This Chapter</th>
<th>Page #</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complete the configuration of the component product or maintenance</td>
<td>☐ &quot;Checklist: Prerequisites for Configuring the Component Product&quot;</td>
<td>page 23</td>
</tr>
<tr>
<td></td>
<td>☐ &quot;Checklist: Setting up the Configuration Environment&quot;</td>
<td>page 24</td>
</tr>
<tr>
<td></td>
<td>☐ &quot;Checklist: Setting up a Runtime Environment and Configuring the Component Product in It&quot;</td>
<td>page 25</td>
</tr>
<tr>
<td></td>
<td>☐ &quot;Checklist: Configuring IBM Tivoli OMEGAMON XE for WebSphere MQ Monitoring in a Runtime Environment&quot;</td>
<td>page 35</td>
</tr>
<tr>
<td></td>
<td>☐ &quot;Checklist: Configuring IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration in a Runtime Environment&quot;</td>
<td>page 31</td>
</tr>
<tr>
<td></td>
<td>☐ “Checklist: Configuring IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers in a Runtime Environment”</td>
<td>page 38</td>
</tr>
<tr>
<td></td>
<td>☐ &quot;Checklist: Configuring IBM Tivoli OMEGAMON XE for WebSphere MQ outside the Configuration Tool”</td>
<td>page 41</td>
</tr>
<tr>
<td></td>
<td>☐ &quot;Checklist: Configuring IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers outside the Configuration Tool”</td>
<td>page 43</td>
</tr>
<tr>
<td></td>
<td>☐ &quot;Checklist: Updating and Changing the Configuration”</td>
<td>page 45</td>
</tr>
<tr>
<td></td>
<td>☐ &quot;Checklist: Using Advanced Features Available in the Configuration Tool”</td>
<td>page 46</td>
</tr>
<tr>
<td>Update or change the configuration for the component product</td>
<td>☐ &quot;Checklist: Setting up the Configuration Environment”</td>
<td>page 24</td>
</tr>
<tr>
<td></td>
<td>☐ &quot;Checklist: Updating and Changing the Configuration”</td>
<td>page 45</td>
</tr>
<tr>
<td></td>
<td>☐ &quot;Checklist: Using Advanced Features Available in the Configuration Tool”</td>
<td>page 46</td>
</tr>
</tbody>
</table>
Checklist: Prerequisites for Configuring the Component Product

This checklist contains the prerequisite actions you must have completed before you can begin to configure the component product:

### Table 3. Prerequisites for Configuring the Component Product

<table>
<thead>
<tr>
<th>✓</th>
<th>Prerequisite</th>
<th>Location of the Information You Will Need</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>If you have not already done so, verify for the component product you want to configure the following:</td>
<td>▪ Program Directory</td>
</tr>
<tr>
<td></td>
<td>□ You have the required software.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>□ You have the required DASD space.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>□ The component product versions can exist in the same Consolidated Software Inventory (CSI).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>You must have installed the component product using the System Modification Program/Extended (SMP/E) tool.</td>
<td>▪ Program Directory</td>
</tr>
<tr>
<td></td>
<td>You must have completed the configuration for a Candle Management Server (CMS).</td>
<td>▪ Configuring Candle Management Server on z/OS</td>
</tr>
<tr>
<td></td>
<td>If this is the first time you have used IBM products or you are not familiar with the terminology or with the Configuration tool, read the overview.</td>
<td>▪ “Overview of Processes, Methods, and the Configuration Tool Modes” on page 47</td>
</tr>
<tr>
<td></td>
<td>If this is the first time you have used IBM products or you do not know the configuration you want, review the background, planning information, and migration information (if any).</td>
<td>▪ “Planning the Configuration” on page 53</td>
</tr>
</tbody>
</table>
Checklist: Setting up the Configuration Environment

Checklist

This checklist contains the steps you perform to set up the configuration environment:

Table 4. Setting up the Configuration Environment

<table>
<thead>
<tr>
<th>✓</th>
<th>Task</th>
<th>Location of the Information You Will Need</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>After installing the component product tape that contains the FMID (HKCI310), read the overview, which says that you must copy the contents from one of the target libraries into the appropriate Configuration tool work library. This applies to products installed into either an existing CSI or into a new CSI.</td>
<td>“Setting up the Configuration Environment and Starting the Configuration Tool” on page 76</td>
</tr>
<tr>
<td></td>
<td>If the component products are being installed into an existing CSI, follow the procedure for copying the contents of the target library (&amp;thilev.TKCIINST) to the existing Configuration tool work library (&amp;shilev.INSTLIBW).</td>
<td>“If you use an existing Consolidated Software Inventory” on page 76</td>
</tr>
<tr>
<td></td>
<td>If the component products are being installed into a new CSI, follow the procedure for copying the contents of the target library (&amp;thilev.TKCIINST) to the newly created Configuration tool work library (&amp;shilev.INSTLIB).</td>
<td>“If you use a new Consolidated Software Inventory” on page 77</td>
</tr>
</tbody>
</table>
Checklists: Setting up a Runtime Environment and Configuring the Component Product in It

Checklist: Setting up a runtime environment and configuring the component product in it

Review the following table to determine the checklist or topic appropriate for your needs:

Table 5. Setting up a Runtime Environment and Configuring the Component Product in It

<table>
<thead>
<tr>
<th>Task</th>
<th>Resources to Use in this Topic</th>
<th>Page #s</th>
</tr>
</thead>
<tbody>
<tr>
<td>You want to configure a component product in a new runtime environment</td>
<td>“Checklist: Setting up a new runtime environment and configuring the component product in it”</td>
<td>page 25</td>
</tr>
<tr>
<td>You want to configure a component product in an existing runtime environment</td>
<td>“Checklist: Setting up an existing runtime environment and configuring the component product in it”</td>
<td>page 28</td>
</tr>
</tbody>
</table>

Checklist: Setting up a new runtime environment and configuring the component product in it

Table 6. Setting up a New Runtime Environment and Configuring the Component Product in It

<table>
<thead>
<tr>
<th>✓</th>
<th>Task</th>
<th>Location of the Information You Will Need</th>
</tr>
</thead>
<tbody>
<tr>
<td>Θ</td>
<td>Select Configure products on the Configuration tool Main Menu and complete the following options on the menu:</td>
<td>“Initial Steps for Setting up a Runtime Environment and Configuring the Component Product in it” on page 82</td>
</tr>
<tr>
<td>Θ</td>
<td>Select Set up configuration environment to specify values that the Configuration tool uses to build the Job Control Language (JCL) for the package or component product that you are configuring.</td>
<td></td>
</tr>
<tr>
<td>Θ</td>
<td>Select Select product to configure to display a list of the package or component products available and to specify the package or component product you want to configure.</td>
<td></td>
</tr>
<tr>
<td>☐</td>
<td>Add a runtime environment. Do the following:</td>
<td>“Adding a Runtime Environment” on page 85</td>
</tr>
<tr>
<td>☐</td>
<td>On the Runtime Environments panel, enter A (Add RTE) in the Action field of the first (blank) row in the list.</td>
<td></td>
</tr>
<tr>
<td>☐</td>
<td>Use the remaining panels the Configuration tool displays to specify the values for the runtime environment (such as the name and type).</td>
<td></td>
</tr>
</tbody>
</table>
Table 6. Setting up a New Runtime Environment and Configuring the Component Product in It (continued)

<table>
<thead>
<tr>
<th>Task</th>
<th>Location of the Information You Will Need</th>
</tr>
</thead>
</table>
| ☑ Build the runtime libraries. Do the following:  
  ☐ On the Runtime Environments panel, enter B (Build libraries) in the Action field next to the runtime environment you are configuring.  
  ☐ Review and submit the JCL that allocates the runtime libraries for the component products in the runtime environment. | “Building Runtime Libraries” on page 89 |
Table 6. Setting up a New Runtime Environment and Configuring the Component Product in It (continued)

<table>
<thead>
<tr>
<th>Task</th>
<th>Location of the Information You Will Need</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure the component product in a runtime environment. Do the following:</td>
<td></td>
</tr>
<tr>
<td>On the Runtime Environments panel, enter C (Configure) in the Action field next to the runtime environment you are configuring.</td>
<td>For IBM Tivoli OMEGAMON XE for WebSphere MQ Monitoring, “Checklist: Configuring IBM Tivoli OMEGAMON XE for WebSphere MQ Monitoring in a Runtime Environment” on page 35.</td>
</tr>
<tr>
<td>Use the remaining panels the Configuration tool displays to specify the values for the configuration.</td>
<td>For IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration, “Checklist: Configuring IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration in a Runtime Environment” on page 31.</td>
</tr>
<tr>
<td></td>
<td>For IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers, “Checklist: Configuring IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers in a Runtime Environment” on page 38.</td>
</tr>
</tbody>
</table>
Checklists: Setting up a Runtime Environment and Configuring the Component Product in It

Table 6. Setting up a New Runtime Environment and Configuring the Component Product in It (continued)

<table>
<thead>
<tr>
<th>✓</th>
<th>Task</th>
<th>Location of the Information You Will Need</th>
</tr>
</thead>
<tbody>
<tr>
<td>☐</td>
<td>Load the runtime libraries. Do the following:</td>
<td></td>
</tr>
<tr>
<td>☐</td>
<td>On the Runtime Environments panel, enter L (Load all component product libraries after SMP/E) in the Action field next to the runtime environment you are configuring.</td>
<td></td>
</tr>
<tr>
<td>☐</td>
<td>Review and submit the JCL that loads the libraries from the SMP/E target libraries.</td>
<td></td>
</tr>
<tr>
<td>☐</td>
<td>If desired, replicate or use IBM products on other z/OS images.</td>
<td></td>
</tr>
</tbody>
</table>

“Loading Runtime Libraries” on page 91

“Checklist: Using Advanced Features Available in the Configuration Tool”

Checklist: Setting up an existing runtime environment and configuring the component product in it

Table 7. Setting up an Existing Runtime Environment and Configuring the Component Product in It

<table>
<thead>
<tr>
<th>✓</th>
<th>Task</th>
<th>Location of the Information You Will Need</th>
</tr>
</thead>
<tbody>
<tr>
<td>☐</td>
<td>Select <strong>Configure products</strong> on the Main Menu and complete the following options on the menu:</td>
<td></td>
</tr>
<tr>
<td>☐</td>
<td>Select <strong>Set up configuration environment</strong> to specify values that the Configuration tool will use to build the JCL for the package or component product you are configuring.</td>
<td></td>
</tr>
<tr>
<td>☐</td>
<td>Select <strong>Select product to configure</strong> to display a list of the package or component products available and to specify the package or component product you want to configure.</td>
<td></td>
</tr>
</tbody>
</table>

“Initial Steps for Setting up a Runtime Environment and Configuring the Component Product in it” on page 82
Configure the component product in a runtime environment. Do the following:

- On the Runtime Environments panel, enter C (Configure) in the Action field next to the runtime environment you are configuring.
- Use the remaining panels Configuration tool displays to specify the values for the configuration.

<table>
<thead>
<tr>
<th>Task</th>
<th>Location of the Information You Will Need</th>
</tr>
</thead>
<tbody>
<tr>
<td>- For IBM Tivoli OMEGAMON XE for WebSphere MQ Monitoring, “Checklist: Configuring IBM Tivoli OMEGAMON XE for WebSphere MQ Monitoring in a Runtime Environment” on page 35.</td>
<td></td>
</tr>
<tr>
<td>- For IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration, “Checklist: Configuring IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration in a Runtime Environment” on page 31.</td>
<td></td>
</tr>
<tr>
<td>- For IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers, “Checklist: Configuring IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers in a Runtime Environment” on page 38.</td>
<td></td>
</tr>
</tbody>
</table>
Table 7. Setting up an Existing Runtime Environment and Configuring the Component Product in It (continued)

<table>
<thead>
<tr>
<th>✓</th>
<th>Task</th>
<th>Location of the Information You Will Need</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Load the runtime libraries. Do the following:</td>
<td>“Loading Runtime Libraries” on page 91</td>
</tr>
<tr>
<td></td>
<td>□ On the Runtime Environments panel, enter L (Load all component product libraries after SMP/E) in the Action field next to the runtime environment you are configuring.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>□ Review and submit the JCL that loads the libraries from the SMP/E target libraries.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>□ If desired, replicate or use IBM products on other z/OS images.</td>
<td>“Checklist: Using Advanced Features Available in the Configuration Tool”</td>
</tr>
</tbody>
</table>

Load the runtime libraries. Do the following:

- On the Runtime Environments panel, enter L (Load all component product libraries after SMP/E) in the Action field next to the runtime environment you are configuring.
- Review and submit the JCL that loads the libraries from the SMP/E target libraries.

If desired, replicate or use IBM products on other z/OS images.
Checklist: Configuring IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration in a Runtime Environment

Who should use this checklist

Use this checklist if you are configuring IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration.

Checklist

The following checklist contains an introduction to the steps you perform in the Configuration tool to configure IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration:

Table 8. Configuring IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration in a Runtime Environment

<table>
<thead>
<tr>
<th>Location in the Configuration Tool</th>
<th>Task</th>
<th>Location of the Information You Will Need</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Checkmark] Configure OMEGAMON XE for WebSphere MQ Configuration panel</td>
<td>Select Register with local CMS to enable IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration, by registering it to the local CMS.</td>
<td>“Registering with a Local Candle Management Server” on page 94</td>
</tr>
<tr>
<td>![Checkmark] Configure OMEGAMON XE for WebSphere MQ Configuration panel</td>
<td>Select Configure persistent datastore to specify values for allocating and managing a Persistent Data Store (CT/PDS) for the CMS address space.</td>
<td>“Configuring the Persistent Data Store” on page 97</td>
</tr>
</tbody>
</table>
Configure OMEGAMON XE for WebSphere MQ

Configuration panel

Select Specify configuration parameters to specify the following:
- The WebSphere MQ authorized load library and language library
- Whether you want to collect enhanced audit log information
- The nodename the agent will use to connect to the CMS (default is KMC_DEFAULT)

☐ If you change the agent nodename, you must select Register with local CMS to re-register the component product to the local CMS.

☐ If you are connecting an agent to a remote CMS or a CMS in another runtime environment, and you change the MC nodename, do the following:
  - Select Specify configuration parameters to specify the MC nodename on the remote CMS or CMS on the other runtime environment.
  - Select Register with local CMS to register the component product again with the local CMS.

Table 8. Configuring IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration in a Runtime Environment (continued)

<table>
<thead>
<tr>
<th>✓</th>
<th>Location in the Configuration Tool</th>
<th>Task</th>
<th>Location of the Information You Will Need</th>
</tr>
</thead>
<tbody>
<tr>
<td>☐</td>
<td>Configure OMEGAMON XE for WebSphere MQ Configuration panel</td>
<td>Select Specify configuration parameters to specify the following:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- The WebSphere MQ authorized load library and language library</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Whether you want to collect enhanced audit log information</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- The nodename the agent will use to connect to the CMS (default is KMC_DEFAULT)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>☐ If you change the agent nodename, you must select Register with local CMS to re-register the component product to the local CMS.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>☐ If you are connecting an agent to a remote CMS or a CMS in another runtime environment, and you change the MC nodename, do the following:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Select Specify configuration parameters to specify the MC nodename on the remote CMS or CMS on the other runtime environment.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Select Register with local CMS to register the component product again with the local CMS.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>“Specifying Configuration Parameters” on page 102</td>
<td></td>
</tr>
</tbody>
</table>
Table 8. Configuring IBM Tivoli OMEGamon XE for WebSphere MQ Configuration in a Runtime Environment (continued)

<table>
<thead>
<tr>
<th>Location in the Configuration Tool</th>
<th>Task</th>
<th>Location of the Information You Will Need</th>
</tr>
</thead>
<tbody>
<tr>
<td>□ Configure OMEGamon XE for WebSphere MQ Configuration panel</td>
<td>If you are defining an agent to run in an agent address space, select <strong>Specify Agent address space parameters</strong> to provide information required to create the address space and communicate with the CMS. Do one of the following:</td>
<td>“Configuring an Agent Address Space” on page 110</td>
</tr>
<tr>
<td></td>
<td>□ If this is a new agent to be configured in the agent address space, do the following:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>‒ Specify the configuration values for the agent address space.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>‒ Specify advanced agent configuration values by pressing F5.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>‒ Specify the communication protocols (SNA, IP, IPPPIPE) you want the agent address space to support.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>□ If this is the second or third agent to be configured in the agent address space, on the Agent Address Space Selection panel, do the following:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>‒ To share an existing agent address space, enter S (Select) next to the address space name and then select <strong>Create runtime members</strong> to generate the job that updates the runtime members for the agent address space.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>‒ To add a new agent address space, enter A (Add) in the Action field of the first (blank) row in the list on the Runtime Environments panel and perform the steps for configuring a new agent in the agent address space.</td>
<td></td>
</tr>
<tr>
<td>□ Configure OMEGamon XE for WebSphere MQ Configuration panel</td>
<td>If you are defining an agent to run in an agent address space, select <strong>Create runtime members</strong> to generate the job that creates the runtime members for the agent address space.</td>
<td>“Creating Runtime Members” on page 119</td>
</tr>
<tr>
<td>□ Configure OMEGamon XE for WebSphere MQ Configuration panel</td>
<td>If you are defining an agent to run in the CMS address space, select <strong>Install Agent into local CMS</strong> to generate the job that installs the agent into the specified CMS.</td>
<td>“Installing the Agent into a Local Candle Management Server” on page 118</td>
</tr>
</tbody>
</table>
Determining what to do next

After finishing up with the task of loading all the libraries used by the runtime environment, which is referred to in “Checklists: Setting up a Runtime Environment and Configuring the Component Product in It” on page 25, see “Checklist: Configuring IBM Tivoli OMEGAMON XE for WebSphere MQ outside the Configuration Tool” on page 41.
Checklist: Configuring IBM Tivoli OMEGAMON XE for WebSphere MQ Monitoring in a Runtime Environment

Who should use this checklist

Use this checklist if you are configuring IBM Tivoli OMEGAMON XE for WebSphere MQ Monitoring.

Checklist

The following checklist contains the steps you perform in the Configuration tool to configure IBM Tivoli OMEGAMON XE for WebSphere MQ Monitoring:

Table 9. Configuring IBM Tivoli OMEGAMON XE for WebSphere MQ Monitoring in a Runtime Environment

<table>
<thead>
<tr>
<th>Task</th>
<th>Location of the Information You Will Need</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select <strong>Register with local CMS</strong> to enable IBM Tivoli OMEGAMON XE for WebSphere MQ Monitoring, by registering it to the local CMS.</td>
<td>“Registering with a Local Candle Management Server” on page 94</td>
</tr>
</tbody>
</table>
| Select **Specify configuration parameters** to specify the following:  
- The WebSphere MQ authorized load library and language library  
- Monitoring characteristics for the DEFAULT monitoring group | “Specifying Configuration Parameters” on page 102 |
Table 9. Configuring IBM Tivoli OMEGAMON XE for WebSphere MQ Monitoring in a Runtime Environment (continued)

<table>
<thead>
<tr>
<th>Location in the Configuration Tool</th>
<th>Task</th>
<th>Location of the Information You Will Need</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure OMEGAMON XE for WebSphere MQ Monitoring panel</td>
<td>If you are defining an agent to run in an agent address space, select <strong>Specify Agent address space parameters</strong> to provide information required to create the address space and communicate with the CMS. Do on of the following:</td>
<td>▶ “Configuring an Agent Address Space” on page 110</td>
</tr>
<tr>
<td>□</td>
<td>- If this is a new agent to be configured in the agent address space, do the following:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Specify the configuration values for the agent address space.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Specify advanced agent configuration values by pressing F5.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Specify the communication protocols (SNA, IP, IPPIPE) you want the agent address space to support.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>□ If this is the second or third agent to be configured in the agent address space, on the Agent Address Space Selection panel, do the following:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- To share an existing agent address space, enter S (Select) next to the address space name and then select <strong>Create runtime members</strong> to generate the job that updates the runtime members for the agent address space.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- To add a new agent address space, enter A (Add RTE) in the Action field of the first (blank) row in the list on the Runtime Environments panel and perform the steps for configuring a new agent in the agent address space.</td>
<td></td>
</tr>
<tr>
<td>□</td>
<td>If you are defining an agent to run in an agent address space, select <strong>Create runtime members</strong> to generate the job that creates the runtime members for the agent address space.</td>
<td>▶ “Creating Runtime Members” on page 119</td>
</tr>
</tbody>
</table>
Table 9. Configuring IBM Tivoli OMEGAMON XE for WebSphere MQ Monitoring in a Runtime Environment (continued)

<table>
<thead>
<tr>
<th>Location in the Configuration Tool</th>
<th>Task</th>
<th>Location of the Information You Will Need</th>
</tr>
</thead>
</table>
| Configure OMEGAMON XE for WebSphere MQ Monitoring panel | If you are defining an agent to run in an agent address space, select **Configure persistent datastore** to specify values for allocating and managing a CT/PDS for the agent address space. This option is required if these are true:  
  - You will be enabling historical data collection on CandleNet Portal.  
  - You configured the CT/PDS for a previous version of the WebSphere MQ monitoring agent in this runtime environment. | “Configuring the Persistent Data Store” on page 97 |
| Configure OMEGAMON XE for WebSphere MQ Monitoring panel | If you are defining an agent to run in the CMS address space, select **Install Agent into local CMS** to generate the job that installs the agent into the specified CMS. | “Installing the Agent into a Local Candle Management Server” on page 118 |
| Configure OMEGAMON XE for WebSphere MQ Monitoring panel | If you are defining an agent to run in the CMS address space, select **Configure persistent datastore** to specify values for allocating and managing a CT/PDS for the CMS address space. | “Configuring the Persistent Data Store” on page 97 |
| Configure OMEGAMON XE for WebSphere MQ Monitoring panel | Select **Complete the Configuration** to display a set of instructions for completing the configuration outside of the Configuration tool. | “Checklist: Configuring IBM Tivoli OMEGAMON XE for WebSphere MQ outside the Configuration Tool” on page 41 |

Determining what to do next

After finishing up with the task of loading all the libraries used by the runtime environment, which is referred to in “Checklists: Setting up a Runtime Environment and Configuring the Component Product in It” on page 25, see “Checklist: Configuring IBM Tivoli OMEGAMON XE for WebSphere MQ outside the Configuration Tool” on page 41.
Checklist: Configuring IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers in a Runtime Environment

Who should use this checklist

Use this checklist if you are configuring IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers.

Checklist

The following checklist contains the steps you perform in the Configuration tool to configure IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers.

The steps are listed in the order they must be performed. Use the ✓ column to check off steps as you complete them:

Table 10. Configuring IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers in a Runtime Environment

<table>
<thead>
<tr>
<th>✓</th>
<th>Location in the Configuration Tool</th>
<th>Task</th>
<th>Location of the Information You Will Need</th>
</tr>
</thead>
<tbody>
<tr>
<td>☐</td>
<td>Configure OMEGAMON XE for WebSphere Integration Brokers panel</td>
<td>If the agent will communicate with a local CMS, select Register with local CMS to register the agent with the CMS.</td>
<td>“Registering with a Local Candle Management Server” on page 94</td>
</tr>
<tr>
<td>☐</td>
<td>Configure OMEGAMON XE for WebSphere Integration Brokers panel</td>
<td>Select Specify configuration parameters to define a WebSphere Integration Brokers agent.</td>
<td>“Specifying Configuration Parameters” on page 102</td>
</tr>
<tr>
<td>☐</td>
<td>Configure OMEGAMON XE for WebSphere Integration Brokers panel</td>
<td>Select Specify link edit libraries to prompt for site-specific library names that the Configuration tool will need to run the necessary linkedit jobs.</td>
<td>“Specifying Linkedit Libraries for IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers” on page 122</td>
</tr>
<tr>
<td>☐</td>
<td>Configure OMEGAMON XE for WebSphere Integration Brokers panel</td>
<td>Select Create configuration parameters to generate a batch job that will provide the necessary members to create or update the runtime members.</td>
<td>“Creating Configuration Parameters for IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers” on page 123</td>
</tr>
</tbody>
</table>
Table 10. Configuring IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers in a Runtime Environment (continued)

<table>
<thead>
<tr>
<th>Location in the Configuration Tool</th>
<th>Task</th>
<th>Location of the Information You Will Need</th>
</tr>
</thead>
</table>
| Configure OMEGAMON XE for WebSphere Integration Brokers panel | Select Specify Agent address space parameters to provide information required to create the address space and communicate with the CMS. Do the following:  
- Specify the configuration values for the agent address space.  
- Specify advanced agent configuration values by pressing F5.  
- Specify the communication protocols (SNA, IP, IPPPIPE) you want the agent address space to support. | “Configuring an Agent Address Space” on page 110 |
| Configure OMEGAMON XE for WebSphere Integration Brokers panel | Select Create runtime members to generate the job that defines the address space and installs the agent into the address space. | “Creating Runtime Members” on page 119 |
| Configure OMEGAMON XE for WebSphere Integration Brokers panel | If you want to archive the data collected so that you can retrieve it at a later date, select Configure persistent datastore to specify the values for the CT/PDS. | “Configuring the Persistent Data Store” on page 97 |
| Configure OMEGAMON XE for WebSphere Integration Brokers panel | Select Complete the configuration to display a set of instructions for completing the configuration outside of the Configuration tool. Also do the following:  
- Within the Configuration tool, if you answered N to the question of whether you want to include the USS steps in the load job, as described in “Loading Runtime Libraries” on page 91, complete this step:  
Before starting the agent and the monitored brokers, execute &shilev.midlev.RKANSAM (KQIUSSJB). | “Checklist: Configuring IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers outside the Configuration Tool” on page 43 |
Determining what to do next

After finishing up with the task of loading all the libraries used by the runtime environment, which is referred to in “Checklists: Setting up a Runtime Environment and Configuring the Component Product in It” on page 25, see “Checklist: Configuring IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers outside the Configuration Tool” on page 43.
Who should use this checklist
Use this checklist if you are configuring IBM Tivoli OMEGAMON XE for WebSphere MQ.

Checklist
The following checklist contains the manual steps you perform outside of the Configuration tool to further configure IBM Tivoli OMEGAMON XE for WebSphere MQ:

### Table 11. Configuring IBM Tivoli OMEGAMON XE for WebSphere MQ outside the Configuration Tool

<table>
<thead>
<tr>
<th>✓</th>
<th>Task</th>
<th>Location of the Information You Will Need</th>
</tr>
</thead>
<tbody>
<tr>
<td>☐</td>
<td>Copy the started task procedures from <code>rhilev.midlev.RKANSAM</code> to the procedure library (PROCLIB).</td>
<td>“Copying the Procedures for the Started Task” on page 126</td>
</tr>
</tbody>
</table>
| ☐ | Copy VTAM® definitions and vary the major node active. Do the following:  
  - Copy VTAM definitions from `rhilev.midlev.RKANSAM` to the VTAM list library (VTAMLST).  
  - Vary the VTAM major node in VTAMLST active. | “Copying the VTAM Definition and Varying the VTAM Node Active” on page 127 |
| ☐ | APF-authorize the RKANMODn runtime load libraries. | “APF-authorizing Libraries” on page 128 |
| ☐ | If you configured the CT/PDS, copy the CT/PDS maintenance procedures from `rhilev.midlev.RKANSAM` to the procedure library (PROCLIB). | “Copying the Procedures for the Persistent Data Store” on page 129 |
| ☐ | Install product support for the CandleNet Portal, and if needed, for a non-z/OS CMS. | “Installing Product Support for CandleNet Portal and Candle Management Server on a Non-z/OS Machine” on page 130 |
| ☐ | Seed the CMS on a remote machine, if required. | “Seeding the Candle Management Server on a Remote Machine” on page 137 |
| ☐ | For IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration connecting to a UNIX CMS, set up the configuration database. | “Setting up the Configuration Database for Candle Management Server on UNIX” on page 138 |
### Table 11. Configuring IBM Tivoli OMEGAMON XE for WebSphere MQ outside the Configuration Tool (continued)

<table>
<thead>
<tr>
<th>Task</th>
<th>Location of the Information You Will Need</th>
</tr>
</thead>
<tbody>
<tr>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>☐</td>
<td>Authorize the external security program to permit IBM Tivoli OMEGAMON XE for WebSphere MQ Monitoring to access the queue managers.</td>
</tr>
<tr>
<td>☐</td>
<td>“Granting Authorizations to IBM Tivoli OMEGAMON XE for WebSphere MQ” on page 146</td>
</tr>
<tr>
<td>☐</td>
<td>Authorize the external security program to permit IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration to configure the queue managers.</td>
</tr>
<tr>
<td>☐</td>
<td>“Enabling Monitoring of CICS Channels for IBM Tivoli OMEGAMON XE for WebSphere MQ” on page 149</td>
</tr>
<tr>
<td>☐</td>
<td>Enable monitoring of CICS® channels.</td>
</tr>
<tr>
<td>☐</td>
<td>All topics beginning with Multi-level Security in the chapter “Completing the Configuration Outside the Configuration Tool (Specific to Component Products)” on page 145</td>
</tr>
<tr>
<td>☐</td>
<td>Implement multi-level security for the z/OS-based configuration database.</td>
</tr>
<tr>
<td>☐</td>
<td>“Verifying the Configuration: Same Address Space” on page 141</td>
</tr>
<tr>
<td>☐</td>
<td>“Verifying the Configuration: Different Address Space” on page 142</td>
</tr>
<tr>
<td>☐</td>
<td>“Verifying the Configuration: CMS on a Different Platform” on page 143</td>
</tr>
<tr>
<td>☐</td>
<td>Verify the configuration. Do one of the following:</td>
</tr>
<tr>
<td>☐</td>
<td>If IBM Tivoli OMEGAMON XE for WebSphere MQ and the CMS are installed and configured in the same address space, do the following:</td>
</tr>
<tr>
<td>☐</td>
<td>– Vary the CMS VTAM major node active.</td>
</tr>
<tr>
<td>☐</td>
<td>– Start the started task for the CMS.</td>
</tr>
<tr>
<td>☐</td>
<td>If IBM Tivoli OMEGAMON XE for WebSphere MQ and the CMS are installed in different address spaces and the CMS is a local CMS, do the following:</td>
</tr>
<tr>
<td>☐</td>
<td>– Vary the CMS VTAM major node active.</td>
</tr>
<tr>
<td>☐</td>
<td>– Start the started task for the CMS.</td>
</tr>
<tr>
<td>☐</td>
<td>– Start the started tasks for the agent.</td>
</tr>
<tr>
<td>☐</td>
<td>If IBM Tivoli OMEGAMON XE for WebSphere MQ and the CMS are installed in different address spaces and the CMS is a non-local CMS, do the following:</td>
</tr>
<tr>
<td>☐</td>
<td>– Start the CMS on the platform where it is installed.</td>
</tr>
<tr>
<td>☐</td>
<td>– Start the started task for the agent.</td>
</tr>
<tr>
<td>☐</td>
<td>If desired, replicate the component product or use the component product on another z/OS image.</td>
</tr>
<tr>
<td>☐</td>
<td>“Checklist: Using Advanced Features Available in the Configuration Tool”</td>
</tr>
</tbody>
</table>
Who should use this checklist

Use this checklist if you are configuring IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers.

Checklist

The following checklist contains the manual steps you perform outside of the Configuration tool to further configure IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers:

<table>
<thead>
<tr>
<th>✓</th>
<th>Task</th>
<th>Location of the Information You Will Need</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Copy the started task procedures from <code>rhilev.midlev.RKANSAM</code> to the procedure library (PROCLIB).</td>
<td>“Copying the Procedures for the Started Task” on page 126</td>
</tr>
<tr>
<td></td>
<td>Copy VTAM® definitions and vary the major node active. Do the following:</td>
<td>“Copying the VTAM Definition and Varying the VTAM Node Active” on page 127</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Copy VTAM definitions from <code>rhilev.midlev.RKANSAM</code> to the VTAM list library (VTAMLST).</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Vary the VTAM major node in VTAMLST active.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>APF-authorize the RKANMODn runtime load libraries.</td>
<td>“APF-authorizing Libraries” on page 128</td>
</tr>
<tr>
<td></td>
<td>If you configured the CT/PDS, copy the CT/PDS maintenance procedures from <code>rhilev.midlev.RKANSAM</code> to the procedure library (PROCLIB).</td>
<td>“Copying the Procedures for the Persistent Data Store” on page 129</td>
</tr>
<tr>
<td></td>
<td>Install product support for the CandleNet Portal, and if needed, for a non-z/OS CMS.</td>
<td>“Installing Product Support for CandleNet Portal and Candle Management Server on a Non-z/OS Machine” on page 130</td>
</tr>
<tr>
<td></td>
<td>Seed the CMS on a remote machine, if required.</td>
<td>“Seeding the Candle Management Server on a Remote Machine” on page 137</td>
</tr>
<tr>
<td></td>
<td>Authorize IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers to receive correct broker data.</td>
<td>“Authorizing IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers” on page 161</td>
</tr>
</tbody>
</table>
Table 12. Configuring IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers outside the Configuration Tool (continued)

<table>
<thead>
<tr>
<th></th>
<th>Task</th>
<th>Location of the Information You Will Need</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Customize the parameters for IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers.</td>
<td>“Setting Parameters for IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers” on page 165</td>
</tr>
<tr>
<td></td>
<td>Install the CandleMonitor Node for IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers.</td>
<td>“Installing the CandleMonitor Node in Broker Environments” on page 163</td>
</tr>
<tr>
<td></td>
<td>Verify the configuration. Do one of the following:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>□ If IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers and the CMS are installed in different address spaces and the CMS is a local CMS, do the following:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>– Vary the CMS VTAM major node active.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>– Start the started task for the CMS.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>– Start the started tasks for the agent.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>□ If IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers and the CMS are installed in different address spaces and the CMS is a non-local CMS, do the following:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>– Start the CMS on the platform where it is installed.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>– Start the started task for the agent.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>□ “Verifying the Configuration: Different Address Space” on page 142</td>
<td></td>
</tr>
<tr>
<td></td>
<td>□ “Verifying the Configuration: CMS on a Different Platform” on page 143</td>
<td></td>
</tr>
</tbody>
</table>
Checklist: Updating and Changing the Configuration

Checklist

In some cases, you will need to update the configuration for a component product.

The following checklist contains the steps you perform inside and outside of the Configuration tool to update the configuration:

Table 13. Updating and Changing the Configuration

<table>
<thead>
<tr>
<th>Task</th>
<th>Location of the Information You Will Need</th>
</tr>
</thead>
</table>
| ![checkmark] Make the changes you want to the configuration in the Configuration tool. Also do the following:  
  ![checkmark] On the Runtime Environments panel, enter L (Load all component product libraries after SMP/E) in the Action field next to the runtime environment you are configuring.  
  ![checkmark] Perform the manual steps required to complete configuration outside of the Configuration tool (if any) | “Updating and Changing the Configuration” on page 168 |
| ![checkmark] If desired, replicate the component product or use the component product on another z/OS image. | “Checklist: Using Advanced Features Available in the Configuration Tool” |
Checklist: Using Advanced Features Available in the Configuration Tool

Checklist

Use the advanced features if you have done the following:

- Completed the appropriate checklist for configuring products
- Completed the appropriate checklist for verifying that the configuration is complete:

<table>
<thead>
<tr>
<th>Task</th>
<th>Advanced Feature</th>
<th>Location of the Information You Will Need</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port an existing runtime environment to one or more z/OS images, without reconfiguring the runtime environment for those z/OS images</td>
<td>System Variable Support</td>
<td>“Enabling System Variable Support” on page 169</td>
</tr>
<tr>
<td>Replicate and transport a runtime environment to one or more z/OS images</td>
<td>Batch mode processing</td>
<td>“Batch Mode Processing” on page 179</td>
</tr>
</tbody>
</table>
Overview of Processes, Methods, and the Configuration Tool Modes

Introduction

This chapter provides information about the terminology used in this guide. It describes the commands and functions used in the Configuration tool. Also introduced are the modes (interactive and batch) available for working in the Configuration tool. Review this chapter if you are not familiar with the Configuration tool, with the terminology used, and with the commands and functions.

Chapter Contents

Terminology ................................................................. 48
Configuration Tool Features .............................................. 49
Configuration Tool Modes: Interactive and Batch .................. 51
The following list contains some of the terms used in this guide and defines those terms:

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Modification Program/Extended (SMP/E) tool</td>
<td>Tool used to do the following:</td>
</tr>
<tr>
<td></td>
<td>- Install new products or new versions of products</td>
</tr>
<tr>
<td></td>
<td>- Install maintenance for existing products</td>
</tr>
<tr>
<td>Configuration tool</td>
<td>Tool used to begin the configuration of products</td>
</tr>
<tr>
<td>Component product</td>
<td>Product or feature provided by IBM that is a separate started task</td>
</tr>
<tr>
<td>Configuring</td>
<td>Making a component product operational by completing the configuration of the component product using the Configuration tool</td>
</tr>
<tr>
<td></td>
<td>and completing the manual steps required outside of the Configuration tool</td>
</tr>
<tr>
<td>Cumulative maintenance</td>
<td>Maintenance through a given date that is customer approved</td>
</tr>
<tr>
<td>Customizing</td>
<td>Modifying the defaults for options and settings and other changes that reflect the needs of the site</td>
</tr>
<tr>
<td>Installing</td>
<td>Loading the contents of the IBM package or maintenance and installing a component product</td>
</tr>
<tr>
<td>Migrating</td>
<td>Preserving the customized data so that you can use it in a newer version of the component product</td>
</tr>
<tr>
<td>Preventive maintenance</td>
<td>Maintenance through a given date that is customer approved but not yet included in Cumulative Maintenance</td>
</tr>
<tr>
<td>Runtime environments</td>
<td>A group of runtime libraries that execute IBM products on a z/OS image</td>
</tr>
<tr>
<td>Runtime libraries</td>
<td>Libraries in the runtime environment that are used by the component product when the component product is started</td>
</tr>
<tr>
<td>Target libraries</td>
<td>SMP/E controlled libraries that contain the source from the distribution media</td>
</tr>
</tbody>
</table>
Configuration Tool Features

This section describes the provided defaults, commands, and functions for the Configuration tool.

(Default provided with the Configuration tool)
Whenever possible, the Configuration tool provides defaults for fields and options. These defaults, when provided, should be sufficient to complete the configuration and can be changed to values specific to the site.

(Assistance provided in the Configuration tool)

Messages
Whenever possible, the Configuration tool checks the values you specify and verifies that you have specified the required values. If the Configuration tool detects an error or omission, it displays a short message.

Online help
Online help is available for each panel by pressing F1. If you want help on how to navigate through the online help, press F1 a second time. (The online help uses a different set of keys to perform navigation than those used by the TSO/E interface.)

Display requirements in ISPF
If you are using a 3270 Model 2 (24 x 80) display, you must turn off the pre-defined function (PF) keys so that the Configuration tool panels are not truncated.

To turn off the pre-defined function keys, enter PFSHOW on any command line and press Enter until the function keys no longer appear.

Restrictions

The length of the high-level qualifier for the runtime libraries must be 26 characters or less.

You cannot use the following:

- The ampersand character (&) as input data in the Configuration tool interactive or batch mode
- The ISPF feature for edit recovery

(If the ISPF RECOVERY ON command is entered, edits will produce a recovery error message. Enter the RECOVERY OFF command to suppress the error messages.)
The list that follows shows some of the commands and functions available in the Configuration tool. You can use these commands for navigation purposes and to display information:

### Table 16. Commands and Functions

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advanced key</td>
<td>On some panels for configuring products, displays panels where you can specify specialized values (such as values for the Persistent Data Store (CT/PDS) for a Candle Management Server (CMS))</td>
</tr>
<tr>
<td>End key</td>
<td>Returns to the previous panel</td>
</tr>
<tr>
<td>Enter key</td>
<td>Accepts the values you have specified and displays the next panel in the process</td>
</tr>
<tr>
<td>HELP command</td>
<td>Displays information about a panel or the extended description for a message.</td>
</tr>
<tr>
<td>README command</td>
<td>Displays the README for the current version of the Configuration tool</td>
</tr>
<tr>
<td>README APP command</td>
<td>Displays information about default applids for started tasks and VTAM, and how the Configuration tool processes VTAM applids</td>
</tr>
<tr>
<td>README ERR command</td>
<td>Displays a list of CLIST error codes and descriptions (for both interactive and batch mode)</td>
</tr>
<tr>
<td>README SYS command</td>
<td>Displays a information about system variable support</td>
</tr>
<tr>
<td>UTIL command</td>
<td>Displays the Installation Services and Utilities menu</td>
</tr>
</tbody>
</table>
Configuration Tool Modes: Interactive and Batch

The Configuration tool features the following two modes:

- Interactive mode
- Batch mode

The following table provides a description for each of the two modes:

Table 17. Configuration Tool Modes

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
</table>
| Interactive | ISPF panel-driven facility that assists you in specifying parameters and tailoring jobs for the following:  
  - Configuring new products  
  - Configuring new versions of products |
| Batch | Facility that creates a single batch job that you can use to build, configure, and load a runtime environment. This single job performs all of the same runtime environment processing as the interactive Configuration tool.  
  Batch mode is a simple and useful way of replicating runtime environments to other z/OS systems. See “Batch Mode Processing” on page 179 for detailed information on batch mode processing. |
Configuration Tool Modes: Interactive and Batch
Introduction

This chapter contains information that you should review before configuring component products of the IBM Tivoli OMEGAMON XE for WebSphere Business Integration package and performing maintenance. This chapter does the following:

- Contains background about the component products
- Helps you determine what component products from other IBM Tivoli packages in the component product’s environment need to be installed and configured

Chapter contents

- Introduction to OMEGAMON XE and OMEGAMON DE .............................................54
- Requirements to Review Before Configuring.................................................................60
- Considerations and Requirements for IBM Tivoli OMEGAMON XE for WebSphere MQ. .....62
- Planning for Setting up a Runtime Environment and Configuring the Component Product in It .64
OMEGAMON XE

OMEGAMON XE is a suite of IBM Tivoli products that monitor and manage system and network applications on a variety of platforms. These products keep track of the availability and performance of all parts of your enterprise from one or more designated workstations, and provide reports you can use to track trends and troubleshoot problems.

You can use OMEGAMON XE to do the following:

- Create situations (conditions to test when monitoring)
- Establish performance thresholds, and raise alerts when thresholds are exceeded or values are matched
- Trace the causes leading up to an alert
- Create and send commands to systems in your managed enterprise by means of the Take Action feature
- Create comprehensive reports about system conditions
- Define your own queries, using the attributes from an installed agent or from an ODBC-compliant data source, to monitor conditions of particular interest

Component products of the OMEGAMON XE platform

The client, server, and agent implementation includes the following:

- A client, CandleNet Portal, with a Java-based user interface for viewing and monitoring your enterprise. CandleNet Portal offers two modes of operation: desktop and browser.
- A CandleNet Portal Server, placed between the client and the CMS, that enables retrieval, manipulation, and analysis of data from the agents. The CandleNet Portal Server is the central repository for all user data.
- A Candle Management Server (CMS), which acts as a collection and control point for alerts received from the agents, and collects their performance and availability data. It also serves as a repository for historical data. The CMS runs on z/OS, UNIX, Windows XP Professional Edition, Windows 2000, or Windows 2003 Server.
- Agents installed on the systems or subsystems you want to monitor. These agents collect and distribute data to a CMS.
- (if necessary) A Candle Management Workstation (CMW), required primarily for the maintenance task of removing obsolete managed systems from the user interface. Although the CandleNet Portal client replaces the CMW as the user interface for your IBM Tivoli monitored environment, the CMW still offers some features not otherwise available, such as the Universal Message Console and the Policy Microscope.
OMEGAMON DE

OMEGAMON DE offers a dashboard view of your enterprise. It gives you a single point of control for managing the resources your business-critical applications rely on, including a range of operating systems, servers, databases, mainframes, and Web components. For example, a typical IT network might have a Web server running on Windows, an application server running on UNIX, a database on z/OS, and a transaction processor on CICS® on the mainframe. OMEGAMON DE brings all these views together in a single window, so you can see when any component is not working as expected.

CandleNet Portal

Running on Windows XP Professional Edition, Windows 2000, or Windows 2003 Server, CandleNet Portal is the interface into your OMEGAMON XE products. In the same way you use your browser home page as a starting point for navigating the Internet, you use CandleNet Portal to get a high-level overview of your network environment. One section of the window displays the Navigator, a tree-like view of your monitored network, with alert icons that appear when problems arise. The rest of the window is filled with views pertinent to the chosen item in the Navigator tree. From the top level or from your home workspace, you can navigate to specific locations to check activity and investigate problems.

Two modes of operation

- Desktop mode, whereby the CandleNet Portal client is installed on your workstation and runs as a desktop application.
Browser mode, whereby you can start CandleNet Portal from your browser, at which time the thin client software is downloaded to your system and thereafter only for software updates.

When using CandleNet Portal in browser mode, you can start it from any workstation by entering the web server URL.

**CandleNet Portal components**

CandleNet Portal includes its own server and two types of client interface components. Here is a brief description of the components you can install at your site.

- **CandleNet Portal Server**: The CandleNet Portal Server communicates directly with your hub CMS. Install at least one CandleNet Portal Server in your network to deploy CandleNet Portal.

- **CandleNet Portal browser client interface (automatically installed with CandleNet Portal)**: In your Internet browser, to start CandleNet Portal browser mode, you can enter the URL for a specific CandleNet Portal browser client installed on your Web server.

- **CandleNet Portal desktop client interface**: The installation choice labeled CandleNet Portal Client (Desktop Edition) installs a Java-based graphical user interface on a Windows workstation. Once the desktop client is installed and configured, you can use it to start CandleNet Portal in desktop mode.

**Agents**

The component products of the IBM Tivoli OMEGAMON XE for WebSphere Business Integration package are agents. Agents are the data collectors. The agents monitor systems, subsystems, or applications, collect data, and pass the data to CandleNet Portal or the CMW through the CMS. The agents pass commands from the user to the system, subsystem, or application. An agent interacts with a single system or application and, in most cases, resides on the same machine where the system or application is running.

Types of agents include the following:

- **Monitoring agents**
  
  These agents collect performance and analysis data for many systems (such as UNIX), subsystems (such as WebSphere), and applications (such as R/3).

- **Alert adapter**
  
  These agents monitor non-IBM-Tivoli monitoring products for a remote system, subsystem, or application, and relay alert information to the CMS.

  Sources of alerts include console and message logs, network-management products, and system-management products. An alert adapter also may have an alert emitter feature that can export IBM Tivoli alerts to a non-IBM-Tivoli monitoring product.

- **Alert emitters**
  
  These agents monitor events (that is, exceptions) from any product running under control of the CMS and, if applicable, relay them to the monitored system, subsystem, or application for corrective action.
Gateways

These agents communicate events to a management application running on a supported platform using a network service. Examples include the SNMP gateways, which communicate events to an SNMP management application running on AIX® or Windows.

Agents run on z/OS, UNIX, Windows XP Professional Edition, Windows 2000, Windows 2003 Server, HP NonStop Kernel, and OS/400; however, not all agents are supported on all platforms.

Component products of the IBM Tivoli OMEGAMON XE for WebSphere Business Integration package

For a full description of the component products of the IBM Tivoli OMEGAMON XE for WebSphere Business Integration package that run on z/OS, see the following guides:

- Using IBM Tivoli OMEGAMON XE for WebSphere MQ Monitoring
- Using IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration
- Using IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers

Candle Management Server

The CMS can run as a stand-alone server, or as a remote server in a hierarchy of servers that report to a master server called the hub CMS. A CMS can be installed on UNIX, z/OS, Windows XP Professional Edition, Windows 2000, or Windows 2003 Server.

Hub Candle Management Server

The hub CMS serves as the focal point for managing your environment. The hub CMS may receive data from:

- Agents running on the same or remote systems.
- Other CMSs running as remote servers in a hierarchical configuration.

Depending on the complexity of your environment, the number of agents you install, and the amount of data you choose to collect, a single CMS may be all that you need. Or, you may want to configure a hierarchical set of CMSs where remote CMSs report to a hub CMS to distribute the activity.

Remote Candle Management Servers

If large amounts of network data are to be collected, excessive traffic can be minimized with the installation of remote CMSs which collect data from the agent and forward it to the hub CMS. Each remote CMS must reside on its own machine and have a unique CMS name (node), but the architectures of various remote CMSs may differ from each other as well as from the hub CMS.
The following table describes the CMS types and their relationships to other CMSs and to the component products of the IBM Tivoli OMEGAMON XE for WebSphere Business Integration package (agents):

<table>
<thead>
<tr>
<th>Relationship</th>
<th>CMS Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMS to CMS</td>
<td>Hub</td>
<td>CMS that does the following:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Receives data from agents and one or more remote CMSs in the environment</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Communicates data to and receives commands from an interface</td>
</tr>
<tr>
<td></td>
<td>Remote</td>
<td>CMS that does the following:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Receives data from agents</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Communicates data to the hub CMS only (It does not communicate directly with an interface.)</td>
</tr>
<tr>
<td>CMS to agent</td>
<td>Local</td>
<td>CMS that is installed in the same runtime environment as the agent</td>
</tr>
<tr>
<td></td>
<td>Non-local</td>
<td>CMS that is installed on the following:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ z/OS, but is not installed and configured in the same runtime environment as the agent</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ z/OS, but is not installed and configured in the same Consolidated Software Inventory (CSI) as the agent. (For example, the CMS is installed using a different INSTLIB.)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ A platform other than z/OS (such as UNIX or Windows)</td>
</tr>
</tbody>
</table>
Figure 2. Example Configuration Including a Remote CMS

- Monitoring agent on managed system
- Hub CMS on UNIX
- Remote CMS on z/OS
- CandleNet Portal Server or CMW
Requirements to Review Before Configuring

Installing the prerequisite OMEGAMON Platform components

The following OMEGAMON Platform components must be installed, with support for IBM Tivoli OMEGAMON XE for WebSphere Business Integration (OMEGAMON XE for BI Support), before you attempt to configure the component products of the IBM Tivoli OMEGAMON XE for WebSphere Business Integration package:

- CMS
- CandleNet Portal Server
- At least one CandleNet Portal desktop client

Follow the directions in the following books, which you can find on the OMEGAMON Platform version 360 and CandleNet Portal Documentation CD that accompanied this package:

- Installing and Setting up OMEGAMON Platform and CandleNet Portal on Windows and UNIX
- Configuring Candle Management Server on z/OS

Communication protocol requirements

Before specifying the communication protocols (SNA, IP, IP.PIPE) you want the agent address space to support, review the following requirements.

- If IP is one of the protocols that the agent address space will support, and you are running multiple TCP/IP interfaces or network adapters on the same z/OS image, specify the network interface card.
- If IP.PIPE is one of the protocols that the agent address space will support, specify values for address translation and partition name.
- For the SNA protocol, specify the VTAM applids and network ID.

Using TCP/IP or IP.PIPE

If the site is using TCP/IP, be aware that network services such as NIS, DNS, and the /etc/hosts file should be configured to return the fully qualified hostname of the CMS and the component product. For example:

```
#HostName.ibm.com
```

This minimizes the risk of inconsistent values being returned for the hostname.

**Note:** When the installers reference the TCP/IP protocol suite, this means the IBM Tivoli component product will use UDP (User Datagram Protocol) for connection.

If the site is using IP.PIPE, be aware of the following limitations:

- There can be at most 16 IP.PIPE processes per host.
IP.PIPE uses one, and only one, physical port per process. Port numbers are allocated using a well-known port allocation algorithm. The first process for a host is assigned port 1918, which is the default.

KDC_PORTS is not supported for IP.PIPE.

**Note:** When the installers reference IP.PIPE, this means the IBM Tivoli component product will use IP (Internet Protocol).

### Specifying multiple network interface cards

If you are running multiple TCP/IP interfaces or network adapters on the same z/OS image, you must specify the network interface card that you want the IBM Tivoli application to use. This allows you to direct the IBM Tivoli application to connect to a specific TCP/IP local interface.

**Note:** IBM Tivoli applications that can have network interface cards include the CMS and all stand-alone agents.

You will set the network interface card parameter while specifying the IP and IP.PIPE communication values for the IBM Tivoli application. During configuration, the Configuration tool will then add the KDCB0_HOSTNAME= parameter in the KppENV member (where the variable pp is the product code) of the runtime environment’s RKANPAR library. For example, the CMS member name is KDSENV.

**Note:** If you are using Interlink’s TCPACCESS, you cannot use this variable.

### Using the Inter-user Communication Vehicle interface

If you are using the Inter-user Communication Vehicle (IUCV) interface, you will indicate this while specifying the TCP (IP and IP.PIPE) communication values for the IBM Tivoli application. During configuration, the Configuration tool adds the IUCV keyword in the KLXINTCP member of the runtime environment’s RKANPAR library.

KLXINTCP is the CT/Engine initialization member for the following:

- IBM’s High Performance Native Sockets (HPNS) TCP/IP protocol support
- IUCV interface support
- Interlink V5.2 HPNS TCP/IP protocol support

**Note:** When using IBM’s HPNS TCP/IP protocol support, do not enter Y.
Considerations and Requirements for IBM Tivoli OMEGAMON XE for WebSphere MQ

Who should read this topic
Read this topic if you are configuring IBM Tivoli OMEGAMON XE for WebSphere MQ.

Reviewing considerations and requirements
Before configuring IBM Tivoli OMEGAMON XE for WebSphere MQ, review the following:

- To execute IBM Tivoli OMEGAMON XE for WebSphere MQ, you must have installed WebSphere MQ on every node where you plan to run the monitoring or configuration features.
- Before starting IBM Tivoli OMEGAMON XE for WebSphere MQ Monitoring, the WebSphere MQ default objects, such as SYSTEM.DEFAULT.MODEL.QUEUE, must exist. If they do not exist in the environment, you must create them prior to starting the agent.
- IBM Tivoli OMEGAMON XE for WebSphere MQ Monitoring requires read access to SYS1.PARMLIB(IETFSSNxx) to find the WebSphere MQ subsystems defined.
- You must have created all queue managers; the z/OS-based IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration cannot create or start a queue manager.
- You must prepare queue managers for monitoring. To monitor WebSphere MQ events, use the ALT QMGR command to ensure that the following WebSphere MQ parameters are enabled:
  - INHIBTEV
  - REMOTEEV
  - PERFMEV
  - LOCALE
  - STRTPEV

Setting up component trace for IBM Tivoli OMEGAMON XE for WebSphere MQ Monitoring
Setting up the z/OS component trace environment for IBM Tivoli OMEGAMON XE for WebSphere MQ Monitoring provides the capability to generate diagnostic information that IBM Software Support may require at a later date. IBM recommends that you perform this step now so the z/OS component trace members will be in place when and if you need them.

Copy the KMQATRSS module from thilev.RKANMOD (where the variable thilev is the target dataset high-level qualifier) to the link pack area (LPA). Member KMQATRSS
controls starting and stopping of the z/OS component trace for IBM Tivoli OMEGAMON XE for WebSphere MQ Monitoring.

As shipped, the trace for IBM Tivoli OMEGAMON XE for WebSphere MQ Monitoring is inactive. Do not modify member CTKMQ00 in the rhilev.midlev.RKANSAM library (where the variable rhilev.midlev is the runtime dataset high-level qualifier and a middle-level qualifier) unless requested to do so by IBM Software Support. This member contains z/OS component trace parameters for IBM Tivoli OMEGAMON XE for WebSphere MQ Monitoring.

**Deciding whether to configure the component product in its own agent address space**

IBM recommends that you configure IBM Tivoli OMEGAMON XE for WebSphere MQ in its own agent address space.

An agent address space can contain one or both of the component products of IBM Tivoli OMEGAMON XE for WebSphere MQ but does not contain a CMS. Configuring a component product of IBM Tivoli OMEGAMON XE for WebSphere MQ in an agent address space enables it to report to either a CMS on a distributed platform or to a z/OS-based CMS.

You can optionally configure the component products to share an agent address space. You can then start and stop the features with a single started task name. The first feature configured becomes the owner of the agent address space and the values you specify for it are shared by the other feature configured in that address space. There is no situation where there should be more than one component product of each type on a z/OS image.

**Note:** If a feature shares its address space with the CMS, the feature’s historical data will not be viewable from CandleNet Portal.

**Warning:** IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration does not support the secondary CMS feature. Do not configure IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration in an agent address space that reports to a secondary CMS.
Planning for Setting up a Runtime Environment and Configuring the Component Product in It

Runtime environment terminology

Definitions of runtime environment terms used in this guide

The following list provides definitions for some of the runtime environment terms used in this guide:

Table 19. Runtime Environment Terminology

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base libraries</td>
<td>Runtime libraries that the configuration process does not alter, and that are shareable between systems. These libraries physically exist in a FULL runtime environment or BASE runtime environment, or as System Modification Program/Extended (SMP/E) target libraries (if a runtime environment shares with SMP/E).</td>
</tr>
<tr>
<td>BASE runtime environment</td>
<td>Configuration containing only shareable runtime libraries, that are a subset of the libraries needed to run component products (Therefore, they must be shared by another runtime environment.)</td>
</tr>
<tr>
<td>FULL runtime environment</td>
<td>Configuration containing a full set of dedicated libraries, that consists of BASE libraries and LPAR-specific libraries.</td>
</tr>
<tr>
<td>LPAR-specific libraries</td>
<td>Runtime libraries that are built during the configuration process to run on a specific LPAR. These libraries contain the unique elements required for a particular LPAR and cannot be shared between z/OS images.</td>
</tr>
<tr>
<td>Runtime environment</td>
<td>Configuration containing a logical grouping of component product libraries. There are five types of runtime environments: Full, Base, Sharing Base, Sharing Full, and Sharing SMP/E Target.</td>
</tr>
<tr>
<td>Runtime libraries</td>
<td>Libraries that tasks reference during execution.</td>
</tr>
<tr>
<td>SHARING BASE</td>
<td>Configuration containing LPAR-specific libraries, that references libraries configured in a BASE runtime environment</td>
</tr>
<tr>
<td>SHARING FULL</td>
<td>Configuration containing LPAR-specific libraries, that references libraries configured in a FULL runtime environment</td>
</tr>
<tr>
<td>SHARING SMP/E TARGET</td>
<td>Configuration containing LPAR-specific libraries, that references libraries managed by SMP/E</td>
</tr>
<tr>
<td>System Modification Program/Extended (SMP/E) target libraries</td>
<td>SMP/E-maintained target libraries</td>
</tr>
</tbody>
</table>
Types of runtime environments

A runtime environment comprises a grouping of runtime libraries that are referenced by tasks as they execute upon a z/OS image. These runtime libraries include base libraries and LPAR-specific libraries.

Types of runtime environments and configurations

The distinction between library types allows you to optimize the component product environment. For example, by allocating common base libraries to a single runtime environment that can be shared by other runtime environments, you can substantially reduce the amount of disk space required, as well as simplify the application of maintenance across remote z/OS images.

To take advantage of runtime environment sharing relationships, you will need to make a decision on the type and number of runtime environments you will be configuring.

Examples of different types of runtime environment configurations

For examples of different types of runtime environment configurations, see “Examples of possible configurations using runtime environments” on page 68. The way you choose to set up the runtime environments depend on the site requirements, maintenance procedures, and DASD sharing capabilities.

Configuration tool features and enhancements

The following features are available when configuring the component products.

Configuration tool batch mode processing

You can use batch mode processing to quickly create an runtime environment for another z/OS image. For more detailed information, see “Batch Mode Processing” on page 179.

System variable support

The Configuration tool now supports z/OS system symbolics for some of the generated configuration values. For more information, see “Enabling System Variable Support” on page 169 or from any Configuration tool panel enter README SYS.

Additional features and enhancements

The following are some additional features and enhancements:

Table 20. Additional Features and Enhancements for Configuring Runtime Environments

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMS support</td>
<td>SMS support has been added to the Configuration tool dataset allocation for the runtime datasets. For more information, from any Configuration tool panel enter README SMS.</td>
</tr>
<tr>
<td>PDSE support</td>
<td>You can now choose to allocate PDSE datasets instead of PDS datasets for runtime libraries that are not load libraries. For more information, from any Configuration tool panel enter README SMS.</td>
</tr>
</tbody>
</table>
Table 20. Additional Features and Enhancements for Configuring Runtime Environments (continued)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Runtime environment load optimization</td>
<td>You can now define an runtime environment to load only changed members to runtime datasets after you apply maintenance or configure component products.</td>
</tr>
<tr>
<td>JES3 support</td>
<td>Configuration jobs are now JES3 compatible.</td>
</tr>
<tr>
<td>Component upgrade alert</td>
<td>During configuration, the Configuration tool will compare the component products being configured to the component products in all runtime environments. If the runtime environment contains a component product at a lower version, an upgrade panel will alert you that an upgrade of these runtime environments is required.</td>
</tr>
<tr>
<td>VTAM applid considerations</td>
<td>For information on VTAM applids, from any Configuration tool panel enter README APP.</td>
</tr>
</tbody>
</table>

Configuration recommendations

Recommended naming convention for VTAM applids

We recommend that you use the default names provided, which use the format ccccppxx, where the following are the variables:

- ccc = 1–4 character global VTAM applid prefix (The default is CTD.)
- pp = 2-character product code
- xx = 2-character specific applid suffix

Enter README APP on the Configuration tool command line to display an extended explanation of how the tool processes applids and lists of default applid names, product codes, and applid suffix values.

The following are some VTAM applid considerations you should be aware of when configuring the component products:

- Applids are standardized across component products and used to communicate between applications.
- VTAM applid prefix and virtual terminal prefix can be specified separately.

Recommended naming conventions for started tasks

We recommend that you use the component product-provided default names, which use the format ccccpp, where the following are the variables:

- cccc = 1–4 character global started task prefix (The default is CANS.)
- pp = 2-character product code

Started task names are standardized across component products.

Enter README APP on a Configuration tool command line to display the list of product codes.
TCP-related configuration requirement

Review the following TCP-related requirement before you configure the component products:

Started task authorization for TCP/IP privileges

If you are using TCP/IP 3.4 and above, to enable the application to connect to TCP/IP you must create an OMVS segment for Resource Access Control Facility (RACF®), or equivalent security system, identifying the CANSpp started task as a Superuser. Enter the following:

\[ \text{ALU CANSpp OMVS(UID(0) HOME(/) PROGRAM(/BIN/SH))} \]

where the variable \( pp \) is the product code.

Special considerations when sharing a BASE runtime environment

For runtime environments that share a common BASE runtime environment, the sample VTPOOL definition statements provided in the RKANSAM(KOBVTPL) member of each SHARING runtime environment will be shared across all runtime environments that share the BASE runtime environment.

Preview of actions to perform

After installing component products and maintenance packages, you will need to perform one or more actions to configure the component products in the runtime environments.

The following table shows the actions available on the Runtime Environments panel. The table shows the actions to use for specific installation and maintenance tasks. The actions are listed in the order they are used, from left to right:

| Table 21. Actions Available from the Runtime Environments Panel |
|------------------|------------------|------------------|------------------|
| **Installation and Maintenance Tasks** | **Add RTE (A)** | **Build libraries (B)** | **Configure (C)** | **Load all component product libraries after SMP/E (L)** |
| You installed new component products or new versions of existing component products and you want to create a new runtime environment. | ✓ | ✓ | ✓ | ✓ |
| You installed new component products or new versions of existing component products and you want to use an existing runtime environment. | ✓ | ✓ | ✓ | ✓ |
Examples of possible configurations using runtime environments

The following five examples show different types of runtime environment configurations. The way you choose to set up the runtime environments will depend on the site requirements, maintenance procedures, and DASD sharing capabilities.

Note: In each example, the field settings and library names are for illustrative purposes only.

Example 1: FULL runtime environment (self-contained)

The FULL runtime environment contains all libraries that are required to execute some IBM Tivoli component products and is the easiest runtime environment to create.

This type of runtime environment can be defined in any situation, but must be defined if the following are true:

- The installation comprises only a single z/OS image.
- You want each of the z/OS images within the installation to be independent.
- There is no DASD sharing between z/OS images.
- You are creating a runtime environment for a specific combination of component products that does not exist in any other runtime environment.

The following example represents a FULL runtime environment called RTE1, that is completely self-contained. All base libraries and LPAR-specific libraries have been allocated within RTE1:

Table 22. FULL Runtime Environment Example

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTE Name</td>
<td>RTE1</td>
</tr>
</tbody>
</table>

Table 21. Actions Available from the Runtime Environments Panel

<table>
<thead>
<tr>
<th>Installation and Maintenance Tasks</th>
<th>Add RTE (A)</th>
<th>Build libraries (B)</th>
<th>Configure (C)</th>
<th>Load all component product libraries after SMP/E (L)</th>
</tr>
</thead>
<tbody>
<tr>
<td>You installed maintenance for existing component products, and the PSPppvvv file indicates that changes need to be made to the configuration values for the component product.</td>
<td></td>
<td>✔️</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>You installed maintenance for existing component products, and the PSPppvvv file indicates that no changes need to be made to the configuration values for the component product.</td>
<td></td>
<td></td>
<td>✔️</td>
<td></td>
</tr>
</tbody>
</table>
Planning for Setting up a Runtime Environment and Configuring the Component Product in It

The following is an example of an LPAR-specific library DD DSNAME resolution:

\[ \text{//RKANPAR DD DSN=PROD.CAN.RTE1.RKANPAR} \]

The following is an example of a Base library DD DSNAME resolution:

\[ \text{//RKANMODL DD DSN=PROD.CAN.RTE1.RKANMODL} \]

Figure 3. FULL Runtime Environment

<table>
<thead>
<tr>
<th>Type</th>
<th>BASE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hilev</td>
<td>PROD.CAN</td>
</tr>
<tr>
<td>Midlev</td>
<td>RTE1</td>
</tr>
<tr>
<td>Shares with</td>
<td>(none)</td>
</tr>
</tbody>
</table>

Table 22. FULL Runtime Environment Example (continued)

Example 2: BASE Runtime Environment

The BASE runtime environment allocates shareable base libraries only. A BASE runtime environment must be used in conjunction with a SHARING BASE in order to obtain the LPAR-specific libraries that are required to run some IBM Tivoli component products. Further, both the BASE runtime environment and the SHARING BASE must have been defined for the same combination of component products.

A BASE runtime environment is typically used in DASD sharing environments, or when maintenance synchronization across multiple systems is desired. In either case, the BASE runtime environment should be defined on the master z/OS image within the installation. Sharing base libraries avoids unnecessary duplication, saves disk space, and simplifies the application of maintenance to a common point.
The following example represents a BASE runtime environment called RTE2:

Table 23. BASE Runtime Environment Example

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTE Name</td>
<td>RTE2</td>
</tr>
<tr>
<td>Type</td>
<td>BASE</td>
</tr>
<tr>
<td>Hilev</td>
<td>COMMON.BASE</td>
</tr>
<tr>
<td>Midlev</td>
<td>(none)</td>
</tr>
<tr>
<td>Shares with</td>
<td>(none)</td>
</tr>
</tbody>
</table>

The following is an example of an LPAR-specific library DD DSNAME resolution:

*There are no LPAR-specific libraries in a BASE RTE.*

The following is an example of a Base library DD DSNAME resolution:

//RKANMODL DD DSN=COMMON.BASE.RKANMODL

Figure 4. BASE Runtime Environment

RTE2

Base Libraries

Example 3: Runtime Environment SHARING with BASE RTE2 (recommended for sharing)

This configuration is recommended for DASD sharing environments.

Utilizing the BASE runtime environment for common datasets, this runtime environment only contains LPAR-specific libraries. The Configuration tool resolves component product configuration elements to correctly point at the LPAR-specific libraries and the BASE runtime environment libraries as necessary.

The following example represents a SHARING BASE runtime environment called RTE3, which obtains its base library information from the BASE runtime environment RTE2:

Table 24. SHARING BASE Runtime Environment Example

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTE Name</td>
<td>RTE3</td>
</tr>
<tr>
<td>Type</td>
<td>SHARING</td>
</tr>
<tr>
<td>Hilev</td>
<td>SYS.V2000</td>
</tr>
</tbody>
</table>
Planning the Configuration

The following is an example of an LPAR-specific library DD DSNAME resolution:

//RKANPAR DD DSN=SYS.V2000.RTE3.RKANPAR

The following is an example of a Base library DD DSNAME resolution:

//RKANMODL DD DSN=COMMON.BASE.RKANMODL

Figure 5. Runtime Environment SHARING with BASE RTE2

Table 24. SHARING BASE Runtime Environment Example

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Midlev</td>
<td>RTE3</td>
</tr>
<tr>
<td>Shares with</td>
<td>BASE RTE2</td>
</tr>
</tbody>
</table>

The SHARING runtime environment allocates LPAR-specific libraries only, and in this example, obtains its base library information from a FULL runtime environment that contains the same combination of component products.

This configuration can also be used for DASD sharing environments, although the BASE/SHARING pair is the recommended approach. The FULL runtime environment should be defined on the master z/OS image in the installation, with sharing runtime environments being defined on the remaining remote images.

The following example represents a SHARING FULL runtime environment called RTE4, which obtains its base library information from the FULL runtime environment RTE1:

Table 25. SHARING FULL Runtime Environment Example

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTE Name</td>
<td>RTE4</td>
</tr>
<tr>
<td>Type</td>
<td>SHARING</td>
</tr>
<tr>
<td>Hilev</td>
<td>X.Y.Z</td>
</tr>
<tr>
<td>Midlev</td>
<td>RTE4</td>
</tr>
<tr>
<td>Shares with</td>
<td>FULL RTE1</td>
</tr>
</tbody>
</table>

Example 4: Runtime environment SHARING with FULL RTE1

The SHARING runtime environment allocates LPAR-specific libraries only, and in this example, obtains its base library information from a FULL runtime environment that contains the same combination of component products.

This configuration can also be used for DASD sharing environments, although the BASE/SHARING pair is the recommended approach. The FULL runtime environment should be defined on the master z/OS image in the installation, with sharing runtime environments being defined on the remaining remote images.

The following example represents a SHARING FULL runtime environment called RTE4, which obtains its base library information from the FULL runtime environment RTE1:
The following is an example of an LPAR-specific library DD DSNAME resolution:

```
//RKANPAR DD DSN=X.Y.Z.RTE4.RKANPAR
```

The following is an example of a Base library DD DSNAME resolution:

```
//RKANMODL DD DSN=PROD.CAN.RTE1.RKANMODL
```

Figure 6. Runtime Environment SHARING with FULL Runtime Environment

Example 5: Runtime environment SHARING with System Modification Program/Extended target libraries

The SHARING runtime environment allocates LPAR-specific libraries only, and in this example, obtains its base library information from target libraries managed by SMP/E.

This configuration is useful for sites that do the following:

- work with limited DASD space, as this configuration method will not allocate base libraries in the runtime environment, thereby reducing storage requirements
- want to immediately activate any SMP/E applied maintenance

The following example represents a SHARING SMP/E TARGET runtime environment called RTE5, which obtains its base library information from SMP/E target libraries:

Table 26. SHARING SMP/E TARGET Runtime Environment Example

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTE Name</td>
<td>RTE5</td>
</tr>
<tr>
<td>Type</td>
<td>SHARING</td>
</tr>
<tr>
<td>Hilev</td>
<td>SYS2.OS26</td>
</tr>
</tbody>
</table>
The following is an example of an LPAR-specific library DD DSNAME resolution:

```
//RKANPAR DD DSN=SYS2.OS26.RTE5.RKANPAR
```

The following is an example of a Base library DD DSNAME resolution:

```
//RKANMODL DD DSN=INSTALL.SMPE.TKANMODL
```

**Figure 7. Runtime Environment SHARING with SMP/E Target Libraries**

**Table 26. SHARING SMP/E TARGET Runtime Environment Example**

<table>
<thead>
<tr>
<th>Midlev</th>
<th>RTE5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shares with</td>
<td>SMP/E Target Libraries</td>
</tr>
<tr>
<td>Hilev (SMP)</td>
<td>INSTALL.SMPE</td>
</tr>
</tbody>
</table>
Planning for Setting up a Runtime Environment and Configuring the Component Product in It

Configuring IBM Tivoli OMEGAMON XE for WebSphere Business Integration on z/OS
Setting up the Configuration Environment and Starting the Configuration Tool

Introduction

This chapter contains topics that correspond to the tasks in the section in Chapter 1 called “Checklist: Setting up the Configuration Environment” on page 24.

Chapter Contents

Setting up the Configuration Environment and Starting the Configuration Tool ............... 76
Setting up the Configuration Environment and Starting the Configuration Tool

After installing the component product tape that contains the FMID (HKCI310), as instructed in the Program Directory, you need to set up the configuration environment.

Setting up the environment consists of copying the contents from one of the target libraries into the appropriate Configuration tool work library. This applies to products installed into either an existing Consolidated Software Inventory (CSI) or into a new CSI. The CSI is a VSAM data set in which System Modification Program/Extended (SMP/E) maintains information about the system.

If you use an existing CSI, you copy the contents of the target library (&thilev.TKCIINST) into the existing Configuration tool work library (&shilev.INSTLIBW). If you use a new CSI, you copy the contents of the target library (&thilev.TKCIINST) to the newly created Configuration tool library (&shilev.INSTLIB).

If you use an existing Consolidated Software Inventory

If you use an existing CSI, perform these steps to copy the contents of the target library into the existing Configuration tool work library and to configure the environment:

1. Copy the contents of the &thilev.TKCIINST library to the &shilev.INSTLIBW library. Enter the following:

```
//COPY EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=*  
//IN DD DSN=&thilev.TKCIINST,DISP=SHR
//OUT DD DSN=&shilev.INSTLIBW,DISP=SHR
/SYSIN DD *
   C O=OUT,I=( (IN,R) )
```

where the following are the variables:

&thilev = the target dataset high-level qualifier
&shilev = the SMP/E dataset high-level qualifier

Start the Configuration tool. The Configuration tool will automatically perform any required updates.

To start the Configuration tool, do the following:

1. Log onto a TSO session.
2. Invoke ISPF.
3. Go to a TSO command line. (In most cases, this is done by entering 6 on the ISPF Primary Option Menu.)
4. Enter the following command:

```
EX '&shilev.INSTLIB'
```

where the variable &shilev is the SMP/E dataset high-level qualifier.

The Configuration tool first displays the copyright panel and then the Main Menu. (Both the copyright panel and the Main Menu display the version and release of the Configuration tool.)
Note: Options related to installation appear in the Main Menu. For the component products of the IBM Tivoli OMEGAMON XE for WebSphere Business Integration package, do not try to access any of the options related to installation that appear in the Main Menu, because installation is done entirely through SMP/E.

2. Perform the initial steps for setting up a runtime environment, starting at “Selecting a package or component product to configure” on page 83.

If you use a new Consolidated Software Inventory

For a new CSI, perform the following steps to copy the contents of the target library to the newly created Configuration tool library, and to set up and configure the environment:

1. Create the &shilev.INSTLIB (where the variable &shilev is the SMP/E dataset high-level qualifier) with the following values:
   
   | RECFM | FB     |
   | LRECL | 80     |
   | BLKSIZE | 8880 |
   | PRIMARY | 600   |
   | SECONDARY | 300  |
   | DIRECTORIES | 88   |

2. Install the package as instructed in the Program Directory.

3. Copy the contents of the &thilev.TKCIINST library into the &shilev.INSTLIB library. Enter the following:

   ```bash
   //COPY  EXEC  PGM=IEBCOPY
   //SYSPRINT DD SYSOUT=*  
   //IN DD DSN=&thilev.TKCIINST,DISP=SHR  
   //OUT DD DSN=&shilev.INSTLIB,DISP=SHR  
   //SYSIN DD *  
   C O=OUT,\( IN,R \) 
   ```

   where the following are the variables:

   - &thilev = the target dataset high-level qualifier
   - &shilev = the SMP/E dataset high-level qualifier

4. Start the Configuration tool. Do the following:

   1. Log onto a TSO session.
   2. Invoke ISPF.
   3. Go to a TSO command line. (In most cases, this is done by entering 6 on the ISPF Primary Option Menu.)
   4. Enter the following command:

   ```bash
   EX '&shilev.INSTLIB'
   ```

   where the variable &shilev is the SMP/E dataset high-level qualifier.

5. From the Main Menu, select Set up work environment and complete the following options on the menu:
1. Select **Specify options** to specify allocation and processing values that will be used to create the work data sets that are needed by the Configuration tool.

2. Select **Allocate work libraries** to allocate the Configuration tool work libraries.

   **Important:** Once you create and submit the Allocate work libraries job, you must exit the Configuration tool and allow the job to run before starting the Configuration tool again.

   **Note:** Options related to installation appear in the Main Menu. For the component products of the IBM Tivoli OMEGAMON XE for WebSphere Business Integration package, do not try to access any of the options related to installation that appear in the Main Menu, because installation is done entirely through SMP/E.

6. Perform the initial steps for setting up a runtime environment, starting at “Initial Steps for Setting up a Runtime Environment and Configuring the Component Product in it” on page 82.
Setting up the Runtime Environment

Introduction

This chapter contains topics that correspond to the tasks in the section in Chapter 1 called “Checklists: Setting up a Runtime Environment and Configuring the Component Product in It” on page 25.

Note: A runtime environment includes the libraries required to execute component products on a z/OS image. One runtime environment is typically configured per z/OS image.

Chapter contents

Overview of Runtime Environment Configuration Sequences ...........................................80
Initial Steps for Setting up a Runtime Environment and Configuring the Component Product in it .................................................................82
Adding a Runtime Environment .....................................................................................85
Building Runtime Libraries ..........................................................................................89
Configuring the Component Product in a Runtime Environment .................................90
Loading Runtime Libraries .........................................................................................91
Deleting a Runtime Environment (optional) ..................................................................92
Overview of Runtime Environment Configuration Sequences

To fully define a runtime environment for the component products, follow the runtime environment configuration sequence below that is appropriate for the situation. Detailed instructions for each action in the configuration sequence are provided in this chapter.

Creating a new runtime environment

Follow this configuration sequence to create a new runtime environment:

1. Add (A) a runtime environment.
   This action defines the runtime environment to the Configuration tool. You are prompted to enter parameters that will be used later to allocate libraries and establish defaults for component product configuration within the runtime environment.

2. Build (B) runtime environment libraries.
   This action generates a batch job that allocates the runtime libraries.

3. (FULL and SHARING runtime environments only) Configure (C) component products in the runtime environment.
   This action collects the parameters required to fully configure a component product within a runtime environment. You cannot configure BASE runtime environments.

4. Load (L) runtime environment libraries.
   This action generates a batch job that loads the runtime environment libraries from the SMP/E target libraries.

   **Note:** The Load action must be performed any time that a component product has been configured within an runtime environment, or if maintenance has been applied.

Adding new component products to an existing runtime environment

Follow this configuration sequence to add new component products to an existing runtime environment:

1. Build (B) runtime environment libraries again.
   This action allocates any additional runtime libraries needed by the new component product. Pre-existing libraries will not be reallocated.

2. (FULL and SHARING runtime environments only) Configure (C) component products in the runtime environment.
   You cannot configure BASE runtime environments.

3. Load (L) runtime environment libraries.

   **Note:** The Load action must be performed any time that a component product has been configured within a runtime environment, or if maintenance has been applied.
Runtime environment configuration sequence for agent address spaces

To create a runtime environment in which one or more agents will reside (without a CMS), follow the runtime environment configuration sequence above that is appropriate for the situation. There is nothing different or special about configuring a runtime environment for this purpose.

Creating a new runtime environment for National Language Support component products

Follow this configuration sequence to create a new runtime environment for National Language Support (NLS) component products:

1. Add (A) a runtime environment.
   Make sure that you enable the “Install National Language components” option on the Add Runtime Environment panel.
2. Build (B) runtime environment libraries.
3. (FULL and SHARING runtime environments only) Configure (C) component products in the runtime environment.
   You cannot configure BASE runtime environments.
4. Load (L) runtime environment libraries.

   Note: The Load action must be performed any time that a component product has been configured within a runtime environment, or if maintenance has been applied.

Adding National Language Support component products to an existing runtime environment

Follow this configuration sequence to add NLS component products to an existing runtime environment:

1. (FULL and SHARING runtime environments only)
   Update (U) a runtime environment to display its configuration information.
   Ensure that you enable the “Install National Language components” option from the Update Runtime Environment panel.
2. Build (B) runtime environment libraries again.
3. (FULL and SHARING runtime environments only)
   Configure (C) component products in the runtime environment to select Japanese VTAM APPLIDs and recreate CT/Engine address spaces.
4. Load (L) runtime environment libraries again.

   Note: The Load action must be performed any time that a component product has been configured within a runtime environment, or if maintenance has been applied.
Initial Steps for Setting up a Runtime Environment and Configuring the Component Product in it

Setting up the runtime environment configuration environment

Before you configure the component products, you may need to specify or update the site-specific configuration environment, such as global DASD information that will be used in any Job Control Language (JCL) that the Configuration tool generates.

Perform the following steps to specify or update the site’s configuration information:

1. From the Main Menu, select Configure products > Set up configuration environment.

2. On the Set Up Configuration Environment panel, specify the high-level qualifiers that identify the attributes used for library allocation for the following library types:

   **Table 27. Set Up Configuration Environment Panel**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VSAM</td>
<td>CSI VSAM Cluster library used by each runtime</td>
</tr>
<tr>
<td>Non-VSAM</td>
<td>Non-VSAM library used by each runtime</td>
</tr>
<tr>
<td>SMP/E Target Datasets</td>
<td>If not provided during component product or maintenance processing, the high-level qualifiers of the SMP/E target libraries must be specified here. Runtime environment creation and maintenance requires access to these libraries. All SMP/E targets must have the same high-level qualifiers if they are to be part of this configuration.</td>
</tr>
</tbody>
</table>

   **Note:** To modify the high-level qualifiers after setting them, you must first unlock them. From the Main Menu, select Services and Utilities > Unlock runtime high-level qualifiers.

3. Perform one of the following procedures:
   - If the runtime libraries are not to be SMS-managed, enter the unit name (Unit) and volume serial number (Volser) to be used for runtime library allocation.
   - If the runtime libraries are to be SMS-managed, do the following:
     - Specify the SMS storage (Storclas) and management class (Mgmtclas) to be used for library allocation.
     - Indicate if you want to use PDSE libraries. PDSEs do not require compression and are not limited by a pre-defined number of directory entries. We recommend that you use PDSEs.

     The default of N indicates that PDS libraries are to be used.

   **Note:** SMS values must be supplied for PDSE libraries.

4. Indicate the technique (RTE allocation routine) you want the Configuration tool to use when generating the runtime environments.
Note: IBM recommends that you use the batch TMP method.

5. After specifying all required site-specific parameters, press Enter.

Viewing configuration information

To view important information about the current release of the Configuration tool, from the Main Menu, select Configure products > Configuration information.

Note: The word “Revised”, when present to the right of this option, indicates that there are updates to this information that you should review.

Managing the environment: configuration services and utilities

To access miscellaneous services and utilities that will help you manage the environment, from the Main Menu, select Configure products > Services and utilities.

Note: Do not modify any values unless you are specifically told to do so in the documentation or by Customer Support personnel. Modifying the Configuration tool values incorrectly may produce unpredictable results or cause the Configuration tool to stop functioning.

See the appendix “Configuration Services and Utilities” on page 211 for details on each of these services and utilities.

Selecting a package or component product to configure

You select a component product for configuration from the Product Selection Menu.

Perform the following steps to select the component product you want to configure:

1. From the Main Menu, select Configure products > Select product to configure.

On the Product Selection Menu, enter S next to the package or component product to select a package or component product to be configured.

Note: Only those packages or component products that are eligible to be configured are listed on this panel. You can only select one product at a time for configuration.

The Runtime Environments panel displays.

2. Perform one of the following procedures next:
   - If you installed new component products or new versions of existing component products, do the following:
     - To create a new runtime environment, see “Adding a Runtime Environment” on page 85
   - If you installed maintenance for existing component products, do the following:
     - If there are configuration changes required, see “Configuring the Component Product in a Runtime Environment” on page 90
Initial Steps for Setting up a Runtime Environment and Configuring the Component Product in it

or

If there are no configuration changes, see “Loading Runtime Libraries” on page 91
Adding a Runtime Environment

Procedure

Perform the following steps to add a new runtime environment:

1. On the Runtime Environments panel enter A (Add RTE) in the Action field of the first (blank) row in the list.

2. Enter a name for the new runtime environment. The name is a unique identifier (up to 8 characters), automatically used as the mid-level qualifier for FULL and SHARING runtime environments. You can optionally specify a mid-level qualifier for BASE runtime environments.

3. Enter the type of runtime environment being created. The following are valid types:

<table>
<thead>
<tr>
<th>Type of Runtime Environment</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FULL</td>
<td>Allocates both private and base libraries. Use this if only one runtime environment will be defined for the environment, or if you add a runtime environment for a unique set of component products.</td>
</tr>
<tr>
<td>BASE</td>
<td>Allocates base libraries only, and does not execute alone. Use this only in conjunction with SHARING runtime environments populated with the same component products.</td>
</tr>
<tr>
<td>SHARING</td>
<td>Allocates private libraries only. This type can share base libraries with a BASE or FULL runtime environment populated with the same component products, or use SMP/E target libraries for its base libraries. Define one SHARING runtime environment for each z/OS image if you have multiple images.</td>
</tr>
</tbody>
</table>

4. (For SHARING runtime environments only) Enter the name of the BASE or FULL runtime environment from which this runtime environment will obtain its base library information. If SMP/E target libraries are to be shared, enter SMP.

5. Enter a description for this runtime environment, which may be any information that is useful for you and others at the site.

6. After specifying all required values on the Runtime Environments panel, press Enter. The first Add Runtime Environment panel displays.

7. Modify the parameters according to the descriptions in the following table:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>

Table 29. First Add Runtime Environment Panel
<table>
<thead>
<tr>
<th><strong>Table 29. First Add Runtime Environment Panel (continued)</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Non-VSAM libraries</strong></td>
</tr>
<tr>
<td>- Enter the high-level qualifier.</td>
</tr>
<tr>
<td>- Enter the site’s values for either the Volser/Unit parameters or the Storclas/Mgmtclas parameters.</td>
</tr>
<tr>
<td>- Indicate whether PDSE libraries are to be used.</td>
</tr>
<tr>
<td>- PDSEs do not require compression, and are not limited by a pre-defined number of directory entries. The default of N signifies that PDS libraries are to be used.</td>
</tr>
<tr>
<td><strong>Note:</strong> Supply SMS values for libraries specified as PDSEs.</td>
</tr>
<tr>
<td><strong>VSAM libraries</strong></td>
</tr>
<tr>
<td>- Enter the high-level qualifier.</td>
</tr>
<tr>
<td>- Enter the site’s values for the Volser or the Storclas/Mgmtclas parameters.</td>
</tr>
<tr>
<td><strong>Mid-level qualifier</strong></td>
</tr>
<tr>
<td>- (For FULL and SHARING runtime environments only) Accept the mid-level qualifier default value (which is the runtime environment name you previously specified) or enter a unique mid-level qualifier.</td>
</tr>
<tr>
<td>- For BASE runtime environments, enter a unique mid-level qualifier or optionally leave this field blank.</td>
</tr>
<tr>
<td><strong>JCL suffix</strong></td>
</tr>
<tr>
<td>- Enter a suffix for the JCL.</td>
</tr>
<tr>
<td>- The suffix (up to four characters) is appended to all JCL that is generated in INSTJOBS. The JCL suffix uniquely identifies the batch job members created by the Configuration tool for this runtime environment.</td>
</tr>
<tr>
<td><strong>STC prefix</strong></td>
</tr>
<tr>
<td>- (For FULL and SHARING runtime environments only) Enter a global STC Prefix (of from 1–4 characters) to be used in building started tasks names for component products in this runtime environment. The default is CANS.</td>
</tr>
<tr>
<td><strong>SYSOUT class / Diagnostic</strong></td>
</tr>
<tr>
<td><strong>SYSOUT class</strong></td>
</tr>
<tr>
<td>- Specify values for the non-diagnostic and diagnostic output DDNAMES.</td>
</tr>
<tr>
<td><strong>Note:</strong> These values were previously hardcoded.</td>
</tr>
<tr>
<td><strong>Load optimization</strong></td>
</tr>
<tr>
<td>- Indicate whether you want to optimize loading of this runtime environment. The default is N.</td>
</tr>
<tr>
<td><strong>Will the runtime environment have a CMS?</strong></td>
</tr>
<tr>
<td>- (For FULL and SHARING runtime environments only) Specify whether a CMS will be configured within this runtime environment.</td>
</tr>
<tr>
<td>- Enter N if no CMS will be configured for this runtime environment. This saves disk space. The default of Y will allocate CMS libraries.</td>
</tr>
<tr>
<td>- You must enter the name of the CMS to be configured; it will be used by other objects that need to communicate with this CMS. The name of the defined runtime environment is used as the default.</td>
</tr>
</tbody>
</table>
Adding a Runtime Environment

8. After specifying all required values on the first Add Runtime Environment panel, press Enter.

The second Add Runtime Environment panel displays.

9. (For FULL and SHARING runtime environments only) Specify values for these parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use OS/390 system variables?</td>
<td>Enter Y if this runtime environment will use z/OS system variables. <strong>Note:</strong> See “Enabling System Variable Support” on page 173 for more information on enabling system variable support.</td>
</tr>
<tr>
<td>Security type</td>
<td>Specify what, if any, security system is to be used for this runtime environment. The default is NONE. If ACF2 is specified, you must also enter the name of the ACF2 macro library.</td>
</tr>
<tr>
<td>VTAM communication values</td>
<td>Enter a global VTAM applid prefix (of from 1–4 characters) to be used in building the VTAM applids for component products in this runtime environment. The default is CTD. Identify the VTAM network. Enter the Logmode table name for LU6.2 logmode entries. The default is KDSMTAB1. Enter the LU6.2 logmode for this runtime environment. The default is CANCTDCS.</td>
</tr>
</tbody>
</table>
10. After specifying all required values on the second Add Runtime Environment panel, press Enter.

**Note:** On the Runtime Environments panel enter *V* (View values) to verify the runtime environment information and *U* (Update) to make the necessary changes.

<table>
<thead>
<tr>
<th>TCP/IP communication values</th>
<th>Note: Enter TSO HOMETEST in the command line to display these values.</th>
</tr>
</thead>
<tbody>
<tr>
<td>▪ Enter the TCP/IP host name of the z/OS system in which this runtime environment resides and the TCP/IP address of the host.</td>
<td></td>
</tr>
<tr>
<td>▪ Enter the started task name of the TCP/IP server.</td>
<td></td>
</tr>
<tr>
<td>▪ Enter the address of the IP port. The default is 1918.</td>
<td></td>
</tr>
<tr>
<td>▪ Enter the Interlink TCP/IP subsystem name (if applicable).</td>
<td></td>
</tr>
</tbody>
</table>
Building Runtime Libraries

Procedure

Perform the following steps to build the runtime libraries:

1. On the Runtime Environments panel enter B (Build libraries) in the Action field of the runtime environment you are configuring.
   A panel displays a JCL for building the runtime environment you are configuring.

2. Review the JCL and submit the job.
   or
   If you want to edit the values in this JCL,
   1. Enter Edit in the command line. The Edit Command Entry panel will display.
   2. Press F1 to access the online help and follow the instructions there.
   3. When the JCL contains the preferred values, press Enter and submit the job.

3. Verify that the job completes successfully. All return codes should be 0.
Now that you have defined at least one runtime environment, you are ready to configure a component product within it. Refer to one of the following checklists for an overview of the configuration tasks required for the component product:

- For IBM Tivoli OMEGAMON XE for WebSphere MQ Monitoring, see “Checklist: Configuring IBM Tivoli OMEGAMON XE for WebSphere MQ Monitoring in a Runtime Environment” on page 35.

- For IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration, see “Checklist: Configuring IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration in a Runtime Environment” on page 31.

- For IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers, see “Checklist: Configuring IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers in a Runtime Environment” on page 38.

These checklists point to procedures specified in the chapter “Configuring the Component Product in a Runtime Environment” on page 93.
Loading Runtime Libraries

When to load runtime libraries

You must load the runtime libraries after you do the following:
- Install and configure the component product in a new runtime environment
- Install and configure an additional component product in an existing runtime environment
- Install maintenance, whether or not you reconfigure a component product

Note: The load job requires exclusive access to the runtime libraries.

Procedure

Perform the following steps to load the runtime libraries from the SMP/E target libraries:

1. On the Runtime Environments panel enter L (Load all product libraries after SMP/E) in the Action field of the runtime environment you are configuring.

   Note: If you are sharing runtime environments, both the master runtime environment and the sharing runtime environment must be loaded.

   A panel displays a JCL for loading the runtime environment you are configuring.

2. If you are configuring IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers, read the instructions in the Load Job - Include USS Steps panel. For the question that asks whether you want to include the USS steps in the load job, enter either Y or N.

   Entering Y to include the USS steps is highly recommended and eliminates the need to manually submit the KQIUSSJB job after the LOAD job completes.

   If N is entered, the KQIUSSJB job will be created and copied into &shilev.midlev.RKANSAM, where the variable &shilev.midlev is the SMP/E dataset high-level qualifier and a middle-level qualifier.

3. Review the JCL and submit the job.

   or

   If you want to edit the values in this JCL,
   1. Enter Edit in the command line. The Edit Command Entry panel will display.
   2. Press F1 to access the online help and follow the instructions there.
   3. When the JCL contains the preferred values, press Enter and submit the job.

4. Verify that the job completes successfully. All return codes should be 0.
Deleting a Runtime Environment (optional)

Considerations when deleting a runtime environment

If you no longer need a particular runtime environment, including the runtime libraries and the configuration values set for the runtime environment, you can delete it. Make sure you really do not need the runtime environment before proceeding. If you delete a runtime environment, any other runtime environment that shares libraries with it will become inoperable.

Be aware that the delete job will delete all libraries that match the following pattern:

\texttt{rhilev.midlev.*}

where the variable \texttt{rhilev.midlev} is the runtime dataset high-level qualifier and a middle-level qualifier. If you have allocated libraries for this runtime environment with a different high-level prefix, then you will have to manually delete those libraries. Also make sure that any libraries that match the pattern \texttt{rhilev.midlev.*} (that may not be part of the runtime environment) are renamed if you do not want to delete those libraries.

As a precaution, we recommend that you back up the entire SMP/E environment and runtime environment, including the INSTLIB, INSTDATA, and INSTJOBS libraries. This allows the runtime environment and INSTLIB to be restored as needed. Restoring only the runtime environment is not sufficient since the runtime environment information is deleted from the INSTLIB.

Procedure

Perform the following steps to delete a runtime environment:

1. On the Runtime Environments panel enter \texttt{D} (Delete) in the Action field of the runtime environment you are configuring.
   The Delete Runtime Environment panel displays.

2. On the Delete Runtime Environment panel, enter \texttt{Y} for Are you sure you want to delete this RTE? and press Enter.
   A panel displays a JCL for deleting the runtime environment.

3. Review the JCL and submit the job.
   or

   If you want to edit the values in this JCL,
   1. Enter \texttt{Edit} in the command line. The Edit Command Entry panel will display.
   2. Press F1 to access the online help and follow the instructions there.
   3. When the JCL contains the preferred values, press Enter and submit the job.
Configuring the Component Product in a Runtime Environment

Introduction

This chapter contains topics that correspond to the tasks in the following sections in Chapter 1:

- “Checklist: Configuring IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration in a Runtime Environment” on page 31
- “Checklist: Configuring IBM Tivoli OMEGAMON XE for WebSphere MQ Monitoring in a Runtime Environment” on page 35
- “Checklist: Configuring IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers in a Runtime Environment” on page 38

Depending on the component products you are configuring, use one or more of these checklists as a guide to configuring the component product in a runtime environment.

After you have performed the procedures in this chapter that apply to the component product, resume the runtime environment setup procedures, beginning with “Loading Runtime Libraries” on page 91.

Chapter contents

Registering with a Local Candle Management Server ........................................94
Configuring the Persistent Data Store .............................................................97
Specifying Configuration Parameters .........................................................102
Configuring an Agent Address Space .........................................................110
Installing the Agent into a Local Candle Management Server .........................118
Creating Runtime Members ....................................................................119
Running the Migration Utility for IBM Tivoli OMEGAMON XE for WebSphere MQ
Configuration (optional) ........................................................................120
Specifying Linkedit Libraries for IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers .........................................................122
Creating Configuration Parameters for IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers ..............................................123
Registering with a Local Candle Management Server

This procedure updates the local Candle Management Server (CMS) to recognize newly installed products. The component products of the IBM Tivoli OMEGAMON XE for WebSphere Business Integration package must be registered with the CMSs to which they report. If a CMS is remote, you must also register the component product with the remote server's corresponding hub CMS.

Note: This option also applies if the component product is running in its own agent address space in another runtime environment and connects to the CMS in this runtime environment.

Procedure

Perform the following procedure to register the component product with a local CMS:

1. On the Runtime Environments panel enter C (Configure) in the Action field of the runtime environment you are configuring.
   The Product Component Selection panel displays.

2. Select the component product you want to configure.
   The Configure OMEGAMON XE panel displays.

3. Select Register with local CMS.

4. If the component product has already been registered, a panel prompts you to confirm a component product refresh. Enter Y or N and press Enter.

5. If you are configuring IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration, the Specify Configuration Parameters panel will appear. Complete the items on the panel, which allow you to select the type of database to use to store the configuration data, using the following procedure:
   1. Enter Internal to use the product-provided database, or enter DB2 to use a version of DB2 Universal Database (DB2) that you have installed.

Note: If you want to use the DB2 type of configuration database, DB2 version 7.1 or higher is required. If DB2 is not already installed on the z/OS machine that hosts the CMS, you can purchase and install it separately, or you can use the Internal type of database.

If you entered Internal, skip to Step “6. Review the Job Control Language (JCL) and submit the job.”
2. Modify the parameters according to the descriptions in the following table:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 subsystem name</td>
<td>The DB2 server subsystem name. If a “DB2 database name” (described below) is specified, the DB2 subsystem name will be placed in the KDS_KCF_DB2_NAME= parameter of RKANPAR(KDSENV). If a “DB2 database name” is not specified, the DB2 subsystem name will be placed in the KDS_KCF_DB2_NAME= parameter of RKANPAR(KCFDBINI).</td>
</tr>
<tr>
<td>DB2 database name</td>
<td>(Optional) Name of the database in which the configuration database information will be created. Maximum of 8 characters. If a value is not specified (left blank), the configuration database information will be automatically created in the DSNDB04 database.</td>
</tr>
<tr>
<td>DB2 user name</td>
<td>The user name for connecting to the DB2 server. Depending upon the site’s configuration and requirements, this may be optional.</td>
</tr>
<tr>
<td>DB2 user password</td>
<td>The password to authenticate the DB2 user. Depending upon the site’s configuration and requirements, this may be optional.</td>
</tr>
<tr>
<td>DB2 library name</td>
<td>The name for the DB2 library</td>
</tr>
</tbody>
</table>

6. Review the Job Control Language (JCL) and submit the job.

or

If you want to edit the values in this JCL,

1. Enter Edit in the command line.
   The Edit Command Entry panel will display.
2. Press F1 to access the online help and follow the instructions there.
3. When the JCL contains the preferred values, press Enter and submit the job.

Changing or migrating the configuration database data for IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration

Migrating configuration database data from a previous installation

If you are keeping the configuration database type as Internal and want to install the configuration database data into the same files as your previous installation of IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration, perform the procedure in “Running the Migration Utility for IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration (optional)” on page 120.

Migrating configuration database data to DB2 from a previous installation

If you are migrating to the DB2 type of configuration database from the configuration database used in a previous installed version, you should have backed up the previous configuration database before installing the new version of IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration. Then perform the following procedure:

1. Configure the type of configuration database as DB2. See “Procedure” on page 94.
2. Run the restore utility to restore the backup into the newly created DB2 database. Read the instructions for restoring a configuration database in the “Backing up the Configuration Database” chapter of *Using IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration*.

**Changing the configuration database type to DB2 after configuring it as Internal**

If you want to change the configuration database type to DB2 after configuring it as Internal, perform the following procedure:

1. Back up the existing configuration database. Read the instructions for backing up a configuration database in the “Backing up the Configuration Database” chapter of *Using IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration*.

2. Configure the type of configuration database as DB2. See “Procedure” on page 94.

3. Run the restore utility to restore the backup into the newly created DB2 database. Read the instructions for restoring a configuration database in the “Backing up the Configuration Database” chapter of *Using IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration*.
Configuring the Persistent Data Store

**Note:** See the appendix called “Maintaining the Persistent Data Store (CT/PDS)” in “Historical Data Collection Guide for IBM Tivoli OMEGAMON XE Products” for an overview of the Persistent Data Store (CT/PDS) facility and the maintenance of it.

Procedure for IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration

**Note:** Any started task associated with a component product (including the CMS address space itself), that is running prior to configuring the CT/PDS, must be stopped.

1. On the Runtime Environments panel enter C (Configure) in the Action field of the runtime environment you are configuring.
   The Product Component Selection panel displays.

2. Select **OMEGAMON XE for WebSphere MQ Configuration**.
   The Configure OMEGAMON XE panel displays.

3. Select **Configure persistent datastore**.
   The Allocate Persistent Data Store panel displays. It introduces the following options:
   1. Modify and review datastore specifications
   2. Create or edit the CT/PDS maintenance jobcard
   3. Create runtime members
   4. Edit and submit datastore allocation job
   5. Complete persistent datastore configuration

4. Select and perform the above procedures in the order they are displayed, according to the following instructions.

**Modifying and reviewing datastore specifications**
Use this panel to control the size, the placement, and the number of datasets that comprise the CT/PDS.
Modify the parameters according to the descriptions in the following table, or press F3 to accept the values displayed:

**Table 32. Persistent Datastore Specifications Panel**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Datasets in Group - Count</td>
<td>The Datasets in Group - Count field specifies the number of datasets that each group contains. The Datasets in Group - Lowlev field is suffixed with a 1-character value (1-9, A-Z) that indicates the dataset number within a group. For example, for a Datasets in Group - Count field value of 3, the MC#PJW job creates the CT/PDS members that make reference to the following datastore files:</td>
</tr>
<tr>
<td></td>
<td>&amp;shilev.midlev.groupname1</td>
</tr>
<tr>
<td></td>
<td>&amp;shilev.midlev.groupname2</td>
</tr>
<tr>
<td></td>
<td>&amp;shilev.midlev.groupname3</td>
</tr>
<tr>
<td></td>
<td>where the variable &amp;shilev.midlev is the SMP/E dataset high-level qualifier and a middle-level qualifier.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Three is the recommended minimum setting for all but a limited number of special purpose datasets. If you enter 1, additional I/O is required to delete records when space is reused. If you enter 2, data becomes unavailable when a datastore file fills up.</td>
</tr>
<tr>
<td>Volser/Unit</td>
<td>The volume serial number and unit name to be used for runtime library allocation</td>
</tr>
<tr>
<td>Storclas/Mgmtclas</td>
<td>The SMS storage class and management class to be used for library allocation</td>
</tr>
<tr>
<td>Est Cyl - Space</td>
<td>The Est Cyl - Space field is used to allocate the number of datastore files you specified for the group. This value is the number of cylinders that will be used for allocating the CT/PDS. The CT/PDS processing computes how much space is needed to allocate the group datastore files, and how much additional required space is needed to hold overhead information. Overhead information includes the product dictionary, table records, index records, and spare room for buffers that need to be reserved for when the datastore file fills up.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Press F5 to invoke a panel to calculate the amount of space required for this product's CT/PDS. This feature is available if this product requires special calculation for its CT/PDS files. Otherwise, specify the number of cylinders you want to use for the datastore files in the -Est Cyl space- field.</td>
</tr>
<tr>
<td>Maintenance - Backup</td>
<td>Enter Y to turn on this maintenance function. The Backup facility is used by customers who want to keep old history data. This facility does an MVS IEBGENER request to make an exact copy of the dataset being maintained. It is a simple copy of the dataset to either tape or DASD. A backed-up file also has the advantage that nothing has to be done to the file to dynamically make the data available to the CT/PDS again.</td>
</tr>
</tbody>
</table>
Configuring the Persistent Data Store

Creating or editing the Persistent Data Store maintenance jobcard

Use this panel to override the jobcard that will be used for the CT/PDS KPDPROC1, KPDPROCC and KPDPROC2 maintenance jobs. Modify the parameter according to the descriptions below, or press F3 to accept the value displayed.

The jobcard member (KPDJOBC) is created in the &shilev.midlev.RKANSAM library, where the variable &shilev.midlev is the SMP/E dataset high-level qualifier and a middle-level qualifier.

This is done via the Create Runtime Members step in the Allocate Persistent Datastore panel. (See “Creating runtime members” on page 100.)

The KPDPROCC exec initiated from KPDPROC1 uses KPDJOBC for running the batch job. All applications that have configured the CT/PDS within this runtime environment will use the same KPDJOBC. The Configuration tool defaults the KPDJOBC setting to the same jobcard that you have specified on the Specify Options panel from the System Modification Program/Extended (SMP/E) installation. The latter jobcard is what was used for the SMP/E jobs.

Note: The CT/PDS maintenance jobs must execute on the same system or image as the applications that support CT/PDS are executing. To do this, you may add a */JOBPARM SYSAFF card or specify the SCHENV= parameter on the job statement depending on the JES configuration. SCHENV is only available on z/OS R4 (and later).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maintenance - Export</td>
<td>Enter Y to turn on this maintenance function. The Export option converts the CT/PDS file to flat files, which makes it easier to process by third party applications. This facility writes the data to a flat file in an internal format. This is also used for recovery purposes when the CT/PDS program detects potential problems with the data. When the data is exported, all of the indexing information is removed from the resulting file, making it a little smaller. The primary use of the Export facility is to perform recovery operations on the dataset. The option to export in the Configuration tool is not needed to perform the recovery operation. To make the exported file available to the CT/PDS program again requires the restore program (KPDREST) to be run against the exported data.</td>
</tr>
<tr>
<td>Maintenance - Extract</td>
<td>Enter Y to turn on this maintenance function. The Extract option goes much further than the export when flattening out the dataset. This facility writes the data to a flat file in readable form which is suitable for loading into other Database Management Systems (DBMS). It will create separate files for each table and convert the internal binary data to EBCDIC. This is a convenient format to load into almost any product. There is no method available to convert the extracted data back to a format that can be used by the CT/PDS.</td>
</tr>
</tbody>
</table>

Note: If none of the above 3 maintenance options is specified (by entering Y), the data within the dataset being maintained will be erased.
Creating runtime members
This panel displays the JCL to allocate parameter members and the procedures needed for the CT/PDSs. Perform one of the following procedures:

- Review the JCL and submit the job.
- If you want to edit the values in this JCL,
  A. Enter Edit in the command line. The Edit Command Entry panel will display.
  B. Press F1 to access the online help and follow the instructions there.
  C. When the JCL contains the preferred values, press Enter and submit the job.

Editing and submitting datastore allocation job
Use this panel to produce the JCL that allocates the CT/PDS. Perform one of the following procedures:

- Review the JCL and submit the job.
- If you want to edit the values in this JCL,
  A. Enter Edit in the command line. The Edit Command Entry panel will display.
  B. Press F1 to access the online help and follow the instructions there.
  C. When the JCL contains the preferred values, press Enter and submit the job.

Completing Persistent Data Store configuration
This selection describes the actions that need to be taken outside the Configuration tool to complete CT/PDS configuration. See “Copying the Procedures for the Persistent Data Store” on page 129.

Procedure for IBM Tivoli OMEGAMON XE for WebSphere MQ Monitoring and IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers

1. On the Runtime Environments panel enter C (Configure) in the Action field of the runtime environment you are configuring.
   The Product Component Selection panel displays.

2. Select the component product you want to configure.
   The Configure OMEGAMON XE panel displays.

3. Select Configure persistent datastore.
   The Configuration tool displays the Specify Persistent Datastore Values panel. Use this panel to specify values for allocating and managing the CT/PDS for use by a standalone
agent. Modify the parameters according to the descriptions in the following table, or press F3 to accept the values displayed:

**Table 33. Specify Persistent Datastore Values Panel**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maintenance procedure prefix</td>
<td>Prefix for the names of the procedures that are used to perform CT/PDS maintenance when a product datastore file is full. Procedure names can be the same for all products in a runtime environment. Specifying the same procedure name prefix for all products is recommended.</td>
</tr>
<tr>
<td>Datastore file high-level prefix</td>
<td>High-level qualifier used in the procedure that allocates CT/PDS files for this standalone agent</td>
</tr>
<tr>
<td>Allocation volume</td>
<td>DASD volume used in the procedure that allocates CT/PDS files for this standalone agent</td>
</tr>
<tr>
<td>Allocation unit</td>
<td>DASD unit used in the procedure that allocates CT/PDS files for this standalone agent. Valid unit types are 3380 and 3390.</td>
</tr>
<tr>
<td>Allocation STORCLAS</td>
<td>SMS storage class to be used in the procedure that allocates CT/PDS files for this standalone agent</td>
</tr>
<tr>
<td>Allocation MGMTCLAS</td>
<td>SMS management class to be used in the procedure that allocates CT/PDS files for this standalone agent</td>
</tr>
</tbody>
</table>
Specifying Configuration Parameters

Procedure for IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration

Perform the following procedure to specify configuration parameters for IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration:

1. On the Runtime Environments panel enter C (Configure) in the Action field of the runtime environment you are configuring. The Product Component Selection panel displays.
2. Select OMEGAMON XE for WebSphere MQ Configuration. The Configure OMEGAMON XE panel displays.
3. Select Specify configuration parameters. The Specify Configuration Parameters panel displays.
4. Modify the parameters according to the descriptions in the following table, or press Enter to accept the values displayed:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM-supplied WebSphere MQ authorized load library: Dataset name</td>
<td>This dataset name applies to the IBM-supplied library which contains the WebSphere MQ API modules that are required to execute IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration. The dataset name you enter here will be added to the STEPLIB DDNAME of the CMS or the agent started task JCL.</td>
</tr>
<tr>
<td>IBM-supplied WebSphere MQ language library: Dataset name</td>
<td>This dataset name applies to the IBM-supplied library that provides National Language Support (NLS). Leave this blank if unavailable.</td>
</tr>
<tr>
<td>Configuration auditing</td>
<td>When you enter Y, configuration auditing will log all configuration changes. Every change will be recorded by saving the before and after state.</td>
</tr>
</tbody>
</table>
Specifying Configuration Parameters

Table 34. Specify Configuration Parameters Panel (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MC agent nodename</td>
<td>This is the nodename the agent will use to connect to the CMS. The value of this field can be the following:</td>
</tr>
<tr>
<td></td>
<td>- KMC_HOSTNAME: The name of the computer that the agent is running on will be used to determine the nodename. For z/OS, the resulting</td>
</tr>
<tr>
<td></td>
<td>nodename will be the current System SMFID.</td>
</tr>
<tr>
<td></td>
<td>- KMC_DEFAULT: The nodename will be determined from the CMS that this agent is connecting to. Based on the KDC_FAMILIES value, the</td>
</tr>
<tr>
<td></td>
<td>resulting nodename will be set in the following ways:</td>
</tr>
<tr>
<td></td>
<td>– SNA: The current System SMFID</td>
</tr>
<tr>
<td></td>
<td>– other: The agent will acquire the nodename value from the CMS. This value is derived using the NCS hostname resolution service</td>
</tr>
<tr>
<td></td>
<td>from either DNS, NIS, or the host file, depending on how the machine is configured.</td>
</tr>
<tr>
<td></td>
<td>- value: The contents of the variable value will be used as the nodename. The nodename is used, along with the concatenated string ::RCACFG,</td>
</tr>
<tr>
<td></td>
<td>to derive the managed system name which will appear in the Managed System folder.</td>
</tr>
</tbody>
</table>

**Note:** If you change the value of the agent nodename and this agent reports to a hub CMS, then you must re-register with the primary hub CMS (local or non-local) that this agent will communicate with. The KMC_NODENAME= parameter value must match in the following RKANPAR members:
- CMS' KDSENV
- Agent's KMCENV or the address space owner's KppENV if IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration is running in a separate address space.

Procedure for IBM Tivoli OMEGAMON XE for WebSphere MQ Monitoring

Perform the following procedure to specify configuration parameters for IBM Tivoli OMEGAMON XE for WebSphere MQ Monitoring:

1. On the Runtime Environments panel enter C (Configure) in the Action field of the runtime environment you are configuring.
   The Product Component Selection panel displays.

2. Select **OMEGAMON XE for WebSphere MQ Monitoring**.
   The Configure OMEGAMON XE panel displays.

3. Select **Specify configuration parameters**.
   The Specify Configuration Parameters panel displays.
4. Modify the parameters according to the descriptions in the following table, or press Enter to accept the values displayed:

**Table 35. Specify Configuration Parameters Panel**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM-supplied WebSphere MQ authorized load library: Dataset name</td>
<td>This dataset name applies to the IBM-supplied library which contains the WebSphere MQ API modules that are required to execute IBM Tivoli OMEGAMON XE for WebSphere MQ Monitoring. The dataset name you enter here will be added to the STEPLIB DDNAME of the CMS or the agent started task JCL.</td>
</tr>
<tr>
<td>IBM-supplied WebSphere MQ language library: Dataset name</td>
<td>This dataset name applies to the IBM-supplied library that provides NLS. Leave this blank if unavailable.</td>
</tr>
<tr>
<td>Sampling interval</td>
<td>This is the IBM Tivoli OMEGAMON XE for WebSphere MQ Monitoring data collection interval (in seconds). The default is <strong>60</strong> seconds.</td>
</tr>
<tr>
<td>Row limit for msgs</td>
<td>This is the maximum number of messages that can be displayed in the message report. If the value is <strong>0</strong>, that means there is no maximum. The default is <strong>0</strong>.</td>
</tr>
<tr>
<td>Values to be set for monitoring group DEFAULT</td>
<td>These values are used to define default settings that the component product will use when no settings are specifically defined for the queue manager. Any of these settings can be overridden at the queue manager level by supplying commands in the KMQUSER member.</td>
</tr>
<tr>
<td>Historical aggregation</td>
<td>The number of history samples to retain in Recent History for all queue managers in the DEFAULT group. The default is <strong>15</strong>.</td>
</tr>
<tr>
<td>Historical display retention</td>
<td>The number of minutes that historical information is retained for queue manager objects (such as channels and queues) that are no longer defined. In most cases, these are queue manager objects that have been deleted. The default is <strong>1440</strong> (24 hours).</td>
</tr>
<tr>
<td>Message access level</td>
<td>The kind of access to allow users to messages in queues. The following are the possible values:</td>
</tr>
<tr>
<td></td>
<td>- <strong>NONE</strong>: No access</td>
</tr>
<tr>
<td></td>
<td>- <strong>DESC</strong>: Allows user to browse the message descriptors. This is the default.</td>
</tr>
<tr>
<td></td>
<td>- <strong>RETRY</strong>: Allows <strong>DESC</strong> plus retry messages from dead letter queue</td>
</tr>
<tr>
<td></td>
<td>- <strong>DATA</strong>: Allows <strong>DESC</strong> and <strong>RETRY</strong> plus lets user browse data portion of messages in queues</td>
</tr>
<tr>
<td></td>
<td>- <strong>DELETE</strong>: Allows <strong>DESC</strong>, <strong>RETRY</strong>, and <strong>DATA</strong> plus lets user delete messages from queues</td>
</tr>
</tbody>
</table>
In the Specify Queue-sharing Group Configuration Parameters panel that displays, modify the parameters according to the descriptions in the following table, or press Enter to accept the values displayed:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| Event queue access | Controls IBM Tivoli OMEGAMON XE for WebSphere MQ Monitoring’s access to WebSphere MQ event queues. The following are the possible values:  
  - **REMOVE**: Allows IBM Tivoli OMEGAMON XE for WebSphere MQ Monitoring to remove messages from the event queues as it processes them. This is the recommended default.  
  - **BROWSE**: Allows IBM Tivoli OMEGAMON XE for WebSphere MQ Monitoring to view the event messages but leaves them in the event queue  
  - **NO**: Tells IBM Tivoli OMEGAMON XE for WebSphere MQ Monitoring to ignore event message processing |
| Command submission enabled | Determines if commands can be submitted to the queue manager via the CMS or CandleNet Portal. The default is **Y**. |
| Candle queue hilevel qualifier | Specifies the high-level qualifier of queue names created by IBM Tivoli OMEGAMON XE for WebSphere MQ Monitoring. The default is **KMQ**. |
| Model queue for reply queues: Queue name | Specifies the name (case sensitive) of the model queue that will be used for reply queues. The default is **SYSTEM.COMMAND.REPLY.MODEL**. |

5. In the Specify Queue-sharing Group Configuration Parameters panel that displays, modify the parameters according to the descriptions in the following table, or press Enter to accept the values displayed:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queue-Sharing Group name</td>
<td>This can specify the name of a particular Queue-Sharing Group, or the default, * (asterisk), to cover all QSGs connected to monitored queue managers.</td>
</tr>
</tbody>
</table>
| QSG monitor | This specifies whether the agent should monitor the given Queue-Sharing Group. The options are the following:  
  - **YES**: The agent will monitor the given Queue-Sharing Group. This is the default.  
  - **NO**: The agent will not monitor the given Queue-Sharing Group.  
  - **TAKEOVER**: The agent will take over monitoring of the given Queue Sharing Group even if another agent is already monitoring it. |
| QSG manager name | This limits the queue managers that are eligible to monitor the QSG. The default, * (asterisk), means any queue manager that is being monitored by the agent may be chosen as the QSG monitoring queue manager. You should take the default unless you need to force the monitoring of QSG to a particular queue manager. |
Procedure for IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers

Perform the following procedure to specify configuration parameters for IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers:

1. On the Runtime Environments panel enter C (Configure) in the Action field of the runtime environment you are configuring.
   The Product Component Selection panel displays.
2. Select OMEGAMON XE for WebSphere Integration Brokers.
   The Configure OMEGAMON XE panel displays.
3. Select Specify configuration parameters.
   The Specify Configuration Parameters panel displays. The parameters in it are required to construct the KQIXML parameter member in RKANDATV.
4. Modify the parameters according to the descriptions in the following table, or press Enter to accept the values displayed on each screen that appears for this panel:

<p>| Table 36. Specify Queue-sharing Group Configuration Parameters Panel (continued) |
|---------------------------------|---------------------------------------------|</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QSG check interval</td>
<td>This specifies the periodic interval (in seconds) at which KMQ QSGAgent does QSG subnode initiation and failover processing. The default is 300 (5 minutes). This parameter is not allowed to be set to less than 60 seconds, with the exception that it can be set to zero to turn off the KMQ QSGAgent thread completely.</td>
</tr>
<tr>
<td>QSG group name</td>
<td>This specifies the alternative Sysplex XCF group name for the coexistence of multiple collection agents. The default is KMQQSG. This parameter is primarily intended for testing purposes, to allow multiple agents to coexist while being tested. Specifying an XCF group name that is in use by other system components could have unexpected consequences and should not be done under any circumstances.</td>
</tr>
</tbody>
</table>

<p>| Table 37. Specify Configuration Parameters Panel |
|---------------------------------|---------------------------------------------|</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HFS CandleHome directory</td>
<td>The name of the HFS CandleHome directory that will be created. There are some product provided modules which will be moved into this directory under different sub-directories. Under the HFS CandleHome directory, sub-directories kqi, kqi/lil, kqi/messages and kqi/messages/En US will also be created. This directory must begin with /. The default is /CandleHome.</td>
</tr>
<tr>
<td>Agent Id</td>
<td>This provides a optional short identifier (maximum of four characters) for the agent. The default is for no agent Id to be assigned, represented by NONE on this panel. <strong>Note:</strong> If the site intends to run more than one KqiAgent on the same host, assign each agent a unique agent Id. If the site intends to run more than one broker with the same name on different hosts, assign each agent that monitors a broker a unique agent Id.</td>
</tr>
</tbody>
</table>
Specifying Configuration Parameters

Table 37. Specify Configuration Parameters Panel (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of broker events to retain</td>
<td>This determines how many broker events to retain per broker 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. The default is 10 events.</td>
</tr>
<tr>
<td>Number of flow events to retain</td>
<td>This determines how many Message Flow Events to retain per broker 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. The default is 10 events.</td>
</tr>
<tr>
<td>Number of product events to retain</td>
<td>This determines the total number of Product 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. The default is 10 events.</td>
</tr>
<tr>
<td>Number of snapshot samples to retain</td>
<td>This applies to monitoring brokers that are Version 5.0 or higher. It gives the minimum number of Snapshot Message Flow Accounting and Statistics samples that will be retained by the agent for viewing in reports. (This is a minimum. More are retained as needed to satisfy requirements for such features as historical data collection.) The default is 15 samples.</td>
</tr>
<tr>
<td>Number of archive samples to retain</td>
<td>This applies to monitoring brokers that are Version 5.0 or higher. It gives the minimum number of Archive Message Flow Accounting and Statistics samples that will be retained by the agent for viewing in reports. (This is a minimum. More are retained as needed to satisfy requirements for such features as historical data collection.) The default is 5 samples.</td>
</tr>
<tr>
<td>Discovery interval</td>
<td>This determines the period (in seconds) between the agent's rediscoveries of brokers created on the system. This parameter can be set to a high value when you are no longer creating new brokers, unless you have a dynamic environment in which new brokers are created frequently. The default is 300 seconds.</td>
</tr>
<tr>
<td>Statistics interval</td>
<td>This determines the minimum period (in seconds) between the collection of broker statistics, such as message flow statistics. The default is 60 seconds.</td>
</tr>
<tr>
<td>Flow event interval</td>
<td>This determines the period (in seconds) for the Message Flow Event sampling interval. This interval does not apply to Broker Events or Product Events. The default is 15 seconds.</td>
</tr>
<tr>
<td>Hold time for query</td>
<td>This gives the number of seconds that the agent will retain data samples for more detailed report viewing after a user has selected a report showing data from the samples. This only applies to reports available in Archive Message Flow Accounting and Snapshot Message Flow Accounting navigator items that are available for brokers that are Version 5.0 or higher. The default is 180 seconds.</td>
</tr>
</tbody>
</table>
Table 37. Specify Configuration Parameters Panel (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| Default take action users     | Specify which CandleNetPortal users are authorized to issue Take Action commands associated with this agent. There can be multiple entries associated with this field, and each entry can be a mask which can include wildcard characters, * (asterisk) and ? (question mark). A CandleNet Portal user whose ID matches any mask in this list is authorized (by the CMS) to issue the Take Action commands that are handled by the KQIAgent. The default is * (asterisk), which allows all CandleNet Portal users to issue Take Action commands associated with this agent.  

**Note:** CandleNet Portal user IDs are defined within the CMS and do not necessarily exist on the node on which the agent is running. This default value applies for any commands that are not associated with a particular broker, and for any broker-related commands for brokers that do not have a specific list of users specified by means of the takeActionAuthUsers parameter. |
| Historical accounting type    | This applies to monitoring brokers that are Version 5.0 or higher. It gives the type of Message Flow Accounting and Statistics data samples that will be logged historically by the agent if historical data collection is active for this data. CandleNet Portal Historical Configuration can be used to activate and deactivate historical data collection. The following are the valid values:  

- **Archive**: Only Archive Message Flow Accounting and Statistics data samples will be logged historically. This is the default.  
- **Snapshot**: Only Snapshot Message Flow Accounting and Statistics data samples will be logged historically.  
- **All**: All types of Message Flow Accounting and Statistics data samples will be logged historically (both Archive and Snapshot).  
- **None**: No Message Flow Accounting and Statistics data samples will be logged historically even if historical data collection is active. |
| Reply queue name              | This specifies the name of the queue that will be used for the agent's receipt of replies and publications for any queue manager to which the agent connects. The queue will be created as a temporary dynamic queue if it does not already exist. If the given name is less than 33 characters long, the actual name of the temporary queue created will be made unique by the queue manager. This value will be used unless a replyQueueName parameter is specifically named for that queue manager. The default is **KQI.AGENT.REPLY.QUEUE**. |
| Reply queue model             | This specifies the name of the queue that will be used as a model for creation of the agent reply queue for any queue manager to which the agent connects. This value will be used unless a replyQueueModel parameter is specifically named for that queue manager. The default is **SYSTEM.BROKER.MODEL.QUEUE**. |
Monitor broker name

This applies to a single broker to be monitored by the agent. One or more of these tags may be specified. If no brokers are specified, no brokers will be monitored. All associated attributes are optional except for the name attribute and the componentDirectory attribute. Optional attributes can be found in *Using IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers*.

Component directory

Enter the directory path created when the broker to be monitored was customized. This is the same as the component directory name required as input to the mqsicreatebroker command with the "-c" parameter. To add an additional broker name and component directory, edit member KQIXML in the RKANDATV dataset. Refer to *Using IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers* for additional information.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monitor broker name</td>
<td>This applies to a single broker to be monitored by the agent. One or more of these tags may be specified. If no brokers are specified, no brokers will be monitored. All associated attributes are optional except for the name attribute and the componentDirectory attribute. Optional attributes can be found in <em>Using IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers</em>.</td>
</tr>
<tr>
<td>Component directory</td>
<td>Enter the directory path created when the broker to be monitored was customized. This is the same as the component directory name required as input to the mqsicreatebroker command with the &quot;-c&quot; parameter. To add an additional broker name and component directory, edit member KQIXML in the RKANDATV dataset. Refer to <em>Using IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers</em> for additional information.</td>
</tr>
</tbody>
</table>
Configuring an Agent Address Space

An agent address space can contain an agent, but does not contain a CMS. Configuring an agent in an agent address space enables it to report to either a CMS on a distributed platform or to a z/OS-based CMS, and is recommended. Each agent can be configured with its own agent address space. You must complete this step to define an agent to run in an agent address space.

Sharing an agent address space

**Note:** The procedure of sharing an agent address space does not apply to the IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers component product.

You can configure IBM Tivoli OMEGAMON XE for WebSphere MQ Monitoring and IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration (no more than one agent of each type) to share an agent address space. You can then start and stop the agents with a single started task name. The first agent configured becomes the owner of the agent address space and the values you specify for it are shared by all agents configured in that address space.

**Notes:** If the agent is already configured to share an agent address space, a message showing the agent address space owner appears in the upper right corner of the Configure OMEGAMON XE panel.

Before you select an agent address space to share, verify that agents in that address space are not running.

Initial Procedure

Perform the following steps to configure an agent address space:

1. On the Runtime Environments panel enter C (Configure) in the Action field of the runtime environment you are configuring.
   The Product Component Selection panel displays.

2. Select the component product you want to configure.
   The Configure OMEGAMON XE panel displays.

3. Select **Specify Agent address space parameters**.
   The panel that displays next depends on which agent is being configured.

4. Consult the following table to determine which steps to take next:

**Note:** The procedure of sharing an agent address space does not apply to the IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers component product. Thus only
Configuring an Agent Address Space

the first row in the table below will apply when configuring IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers.

Table 38. Panels for Configuring an Agent Address Space

<table>
<thead>
<tr>
<th>Agent</th>
<th>Panel that Displays</th>
</tr>
</thead>
<tbody>
<tr>
<td>The first agent to be configured in the agent address space</td>
<td>The Specify Agent Address Space Parameters panel displays. Proceed to “Specifying configuration values” on page 111.</td>
</tr>
</tbody>
</table>
| The second or third agent to be configured in the agent address space | The Agent Address Space Selection panel displays. This panel lists existing agent address spaces and the agents that are already configured to share them. Perform one of the following procedures:  
  | To select an existing agent address space that the agent you are configuring will share, enter S next to the address space name. For more information, refer to “Sharing an agent address space” on page 110.  
  | To add a new agent address space for the agent you are configuring, enter A (Add RTE) and proceed to “Specifying configuration values” on page 111. |

Specifying configuration values

Perform the following steps to specify configuration values:

1. On the Specify Agent Address Space Parameters panel that displays, modify the parameters according to the descriptions in the following table, or press Enter to accept the values displayed:

Table 39. Specify Agent Address Space Parameters Panel

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent started task</td>
<td>The name of the started task PROC for the agent. Note the name; you will use it later in the configuration.</td>
</tr>
<tr>
<td>Connect to CMS in this RTE</td>
<td>Enter Y if you want to connect to a CMS that is already configured in the same runtime environment as the agent. Enter N if you want to connect to a CMS located in another installation library, to a distributed CMS, or if you want to connect to a CMS with a different port number.</td>
</tr>
</tbody>
</table>
| Name of Primary CMS | Select or enter the name of the primary CMS to which the agent will connect. None indicates no selection has been made. You can specify either a remote or hub CMS as the primary CMS for an agent. Perform one of the following procedures:  
  | To select the z/OS-based CMS to which the agent will report, press F10, and select one from the list.  
  | To specify a CMS located in another installation library, to connect to a distributed CMS, or to update the port number of the agent, press F10, then press F5 and fill in the values for the Specify Agent Primary CMS Values panel. |
2. Perform one or more of the following procedures:

- To accept the values you specified and bypass advanced configuration, proceed to “Specifying communication values” on page 112.
- To specify advanced configuration values, proceed to “Advanced configuration (optional)” on page 114.
- To select a different CMS from the one that has been specified for the component product to connect to, proceed to “Changing the CMS connection” on page 114.

**Specifying communication values**

You supply communication values for each protocol that the agent address space may use. From the Specify Agent Address Space Parameters panel, press Enter.

One or more communication protocol panels display next.

**Specifying agent IP.PIPE communication values**

Perform the following step to specify agent IP.PIPE communication values:

If the agent requires IP.PIPE support, on the Specify Agent IP.PIPE Configuration Values panel that displays, modify the parameters according to the descriptions in the following table, or press Enter to accept the values displayed:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hostname</td>
<td>The TCP/IP host name of the machine where the agent resides</td>
</tr>
<tr>
<td>Address</td>
<td>The TCP/IP address of the machine where the agent resides</td>
</tr>
<tr>
<td>Started Task</td>
<td>The TCP/IP address of the machine where the agent resides</td>
</tr>
<tr>
<td>Network interface card (NIC)</td>
<td>If the site is running multiple TCP/IP interfaces or network adapters on the same z/OS image, specify the network interface card that you prefer for the application to use. This value cannot be specified if you are using Interlink for TCP/IP communications.</td>
</tr>
<tr>
<td>Interlink subsystem</td>
<td>If you are using Interlink TCP/IP (Version 4.1 and earlier), then specify the TCP/ACCESS Interlink subsystem name.</td>
</tr>
<tr>
<td>IUCV interface in use?</td>
<td>Indicate whether you are using the Inter-user Communication Vehicle (IUCV) interface. Do not enter Y when using IBM’s HPNS TCP/IP protocol support.</td>
</tr>
<tr>
<td>Address translation</td>
<td>Indicates whether the site uses address translation when communicating across a firewall</td>
</tr>
</tbody>
</table>

**Table 39. Specify Agent Address Space Parameters Panel (continued)**

<table>
<thead>
<tr>
<th>Specify communication protocols in priority sequence</th>
<th>Specify at least one protocol to be used by the agent to communicate with the primary CMS. The possible values are:</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNA</td>
<td>IP</td>
</tr>
<tr>
<td>IP</td>
<td>IP.PIPE</td>
</tr>
</tbody>
</table>
Configuring an Agent Address Space

Specifying agent TCP/IP communication values

Perform the following step to specify agent TCP/IP communication values:

If the agent requires TCP/IP support, on the Specify Agent/IP Configuration Values panel that displays, modify the parameters according to the descriptions in the following table, or press Enter to accept the values displayed:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hostname</td>
<td>The TCP/IP host name of the machine where the agent resides</td>
</tr>
<tr>
<td>Address</td>
<td>The TCP/IP address of the machine where the agent resides</td>
</tr>
<tr>
<td>Started Task</td>
<td>The TCP/IP address of the machine where the agent resides</td>
</tr>
<tr>
<td>Network interface card (NIC)</td>
<td>If the site is running multiple TCP/IP interfaces or network adapters on the same z/OS image, specify the network interface card that you prefer for the application to use. This value cannot be specified if you are using Interlink for TCP/IP communications.</td>
</tr>
<tr>
<td>Interlink subsystem</td>
<td>If you are using Interlink TCP/IP (Version 4.1 and earlier), then specify the TCP/ACCESS Interlink subsystem name.</td>
</tr>
<tr>
<td>IUCV interface in use?</td>
<td>Indicates whether you are using the IUCV interface. Do not enter Y when using IBM’s HPNS TCP/IP protocol support.</td>
</tr>
</tbody>
</table>

Specifying agent SNA communication values

Perform the following steps to specify agent SNA communication values:

1. If the agent requires SNA support, on the Specify Agent SNA Configuration Values panel, specify (or accept the default for) the VTAM applid prefix.
   
   **Note:** There is only one set of VTAM applids per agent address space regardless of how many agents share it.

2. To specify the other VTAM values press F6.

   The Specify VTAM Applid Values panel displays.

3. Modify the parameters according to the descriptions in the following table, or press Enter to accept the values displayed:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| Major node          | Name of VTAM major node that contains all the VTAM applid definitions for this product. This name is also used to activate the VTAM applids for this product.  
   
   **Note:** You will copy VTAM definitions to the VTAM list. |
Advanced configuration (optional)

Perform the following steps to specify advanced configuration values:

1. From the Specify Agent Address Space Parameters panel, press F5. The Specify Advanced Agent Configuration Values panel displays.

2. Modify the parameters according to the descriptions in the following table, or press F3 to accept the values displayed:

### Table 43. Specify Advanced Agent Configuration Values Panel

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable secondary CMS</td>
<td>Enter Y to optionally designate a secondary CMS to which the agent will connect if its primary CMS becomes unavailable. To remove a secondary CMS definition, enter N.</td>
</tr>
<tr>
<td>Name of secondary CMS</td>
<td>Enter the name of the secondary CMS.</td>
</tr>
<tr>
<td>Enable startup console messages</td>
<td>Enter Y if you want a SYSLOG message on the console to indicate when the agent finishes initializing. The default is N.</td>
</tr>
<tr>
<td>Enable WTO messages</td>
<td>Enter Y for this parameter if you want the agent address space to issues WTOs. (WTOs write information and exception condition messages to the operator consoles. Alert messages are written to the consoles.) The default is N.</td>
</tr>
<tr>
<td>Storage detail logging</td>
<td>Set the interval to monitor storage. The default is 60 minutes.</td>
</tr>
<tr>
<td>Flush VSAM buffers</td>
<td>Set the interval to force all deferred VSAM writes to DASD. The default is 30 minutes.</td>
</tr>
<tr>
<td>Minimum extended storage</td>
<td>Specify the minimum extended (virtual) storage the agent acquires.</td>
</tr>
</tbody>
</table>

Changing the CMS connection

1. From the Specify Agent Address Space Parameters panel, press F10. The Communication Selection Panel, which lists the eligible CMSs that have been configured, displays.

See the table “Details about the Candle Management Server Types and Their Relationships” on page 58 for a description of various types of CMS connections.

2. Perform one of the following procedures:
If any of the CMSs listed in the Communication Selection Panel is the CMS that you want the component product to connect to, do the following:

A. Select that CMS and press Enter.

   The Specify Agent Address Space Parameters panel will display with a changed CMS connection.

B. Press F3.

If none of the CMSs listed in the Communication Selection Panel is the CMS that you want the component product to connect to, do the following:

A. Press F5.

   The Specify Agent Primary CMS Values panel displays.

B. Modify the parameters in the panel according to the descriptions in the following table. Depending on the currently configured type of CMS connection, particular panels will display and particular parameters will need to be configured:

Table 44. Specify Agent Primary CMS Values Panel

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMS name</td>
<td>Name of the NODEID of the CMS to which the agent will connect. This name must match the domain name of a non-z/OS CMS or the CMS_NODEID parameter value in the KDSENV member of the RKANPAR library for a z/OS CMS. If the parameter value contains the literal *SMFID, then you must enter the z/OS system's SMFID in place of this literal. This value is always required.</td>
</tr>
<tr>
<td>LU6.2 logmode</td>
<td>Name of the LU6.2 logmode that was defined for use by the CMS. The IBM default name is CANCTDCS.</td>
</tr>
<tr>
<td>Logmode table name</td>
<td>Name of the logmode table that contains the LU6.2 logmode. The IBM default name is KDSMTAB1.</td>
</tr>
<tr>
<td>Local location broker applid</td>
<td>Enter the Local Location Broker applid of the CMS that this agent will be communicating with.</td>
</tr>
<tr>
<td>Network ID</td>
<td>Enter the identifier of the VTAM network where the CMS that this agent is communicating with is running. You can locate this value on the NETID parameter in the VTAMLST startup member ATCSTRnn.</td>
</tr>
<tr>
<td>Hostname</td>
<td>Specify the TCP ID of the z/OS mainframe where the primary CMS to which the agent will connect is running. To get this value, enter TSO HOMETEST in the command line from the system where the primary CMS is running and use the first qualifier of the TCP Hostname.</td>
</tr>
</tbody>
</table>
Configuring an Agent Address Space

116 Configuring IBM Tivoli OMEGAMON XE for WebSphere Business Integration on z/OS

Address

Specify the TCP address of the z/OS mainframe where the primary CMS to which the agent will connect is running. To get this value, enter TSO HOMETEST in the command line from the system where the primary CMS is running. 129.0.131.214 is an example of a CMS address.

Network interface card (NIC)

Specify the network interface card that you prefer for the IBM application to use. During configuration, the Configuration tool will then add the KDCB0_HOSTNAME= parameter in the agent’s ENV member of the RKANPAR library.

Sites that are running multiple TCP/IP interfaces or network adapters on the same z/OS image require this parameter. Setting this parameter allows users to direct the IBM application to connect to a specific TCP/IP local interface. The function of this field is to select a network adapter, by hostname, to be used for input and output. This field is only applicable for networks with multiple interface cards for which a specific output network interface card is required for successful operation. Applications using Interlink’s TCPACCESS cannot use this variable.

IP port number

Specify the IP port number you will be using. If you are not using the default port number 1918, enter the IP port number you will be using. This value is required if the IPPipe port number is not specified.

IPPIPE port number

Specify the IPPIPE port number you will be using. If you are not using the default port number 1918, enter the IPPIPE port number you will be using. This value is required if IP port number is not specified.

Address translation

Enter Y to configure IPPipe support for communication across firewalls using address translation.

Table 44. Specify Agent Primary CMS Values Panel (continued)
Configuring an Agent Address Space

Table 44. Specify Agent Primary CMS Values Panel (continued)

| Partition name | Specify the partition name which identifies this CMS’s location (namespace) relative to the firewalls used for address translation.  
**Note:** The CMS that this agent will connect to must have a corresponding partition reference entry.  
In this task, the partition table is populated and/or modified. The partition table contains labels and associated socket addresses which should be provided by the firewall administrator.  
The labels in the partition table are configured into and used by IBM products on an external network, outside a firewall, during the CMS connection establishment phase. The first part of CMS connection establishment is the lb lookup, which requires the location brokers to return the socket address of the CMS. The partition table is used by the brokers, matching the client’s partition name (KDC_PARTITION=) to the labels in the partition table. On a match, the associated socket address in the partition table is returned to the client outside the firewall. This socket address will be used by the IBM products to traverse the firewall and connect to the CMS.  
Each entry consists of a label or partition name, a protocol (IP for UDP or IPPIPE for TCP), and a hostname or dotted-decimal IP address. The well-known port (hub port) must be authorized by the firewall administrator. If UDP is the protocol configured in the partition table, then a range of (UDP) ports must be authorized by the firewall administrator (in addition to the well-known port). If TCP is the protocol, no additional ports other than the well-known CMS port need be authorized. |
|---|---|

**Note:** If the component product will be connecting to a remote non-z/OS CMS, you must also perform the procedures in “Installing Product Support for CandleNet Portal and Candle Management Server on a Non-z/OS Machine” on page 130.  
If the component product is IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration connecting to a CMS on UNIX, you must also perform the procedure in “Setting up the Configuration Database for Candle Management Server on UNIX” on page 138.
Installing the Agent into a Local Candle Management Server

Procedure

Perform the following steps to display and edit the JCL that installs the agent into the specified CMS for the component product:

1. On the Runtime Environments panel enter C (Configure) in the Action field of the runtime environment you are configuring.
   The Product Component Selection panel displays.
2. Select the component product you want to configure.
   The Configure OMEGAMON XE panel displays.
3. Select **Install the agent into a local CMS**.
   A panel displays a JCL that configures the component product to run in a local CMS address space.
4. Review the JCL and submit the job.
   or
   If you want to edit the values in this JCL,
   1. Enter **Edit** in the command line. The Edit Command Entry panel will display.
   2. Press F1 to access the online help and follow the instructions there.
   3. When the JCL contains the preferred values, press Enter and submit the job.
Creating Runtime Members

Procedure

Perform the following steps to display and edit the JCL that creates the runtime members for the component product:

1. On the Runtime Environments panel enter C (Configure) in the Action field of the runtime environment you are configuring.
   The Product Component Selection panel displays.
2. Select the component product you want to configure.
   The Configure OMEGAMON XE panel displays.
3. Select **Create runtime members**.
   A panel displays a JCL that creates the runtime members for the component product.
4. Review the JCL and submit the job.
   or
   If you want to edit the values in this JCL,
   1. Enter **Edit** in the command line. The Edit Command Entry panel will display.
   2. Press F1 to access the online help and follow the instructions there.
   3. When the JCL contains the preferred values, press Enter and submit the job.
Running the Migration Utility for IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration (optional)

This panel lets you migrate to the current version elements that you have already configured in the previous version of IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration. You need to specify the complete runtime environment high-level qualifiers you used for the previous version libraries.

This step only applies if you are migrating from another installation library. If you are upgrading the same installation library, then you may skip this step.

Who should read this topic

Read this topic if you are configuring IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration.

Complete runtime environment high-level qualifiers

The complete runtime environment high-level qualifier comprises the high-level qualifier and the middle-level qualifier. The following are examples of the high-level qualifier and middle-level qualifier used for a previous version:

- Non-VSAM high-level qualifier: highlevel1
- VSAM high-level qualifier: highlevel2
- Middle-level qualifier: midlevel

For the above example, the following will be the complete Non-VSAM and VSAM high-level qualifiers:

- highlevel1.midlevel
- highlevel2.midlevel

Procedure

Perform the following steps to run the migration utility:

1. Be sure that the system you are migrating from is not running during this process.
2. On the Runtime Environments panel enter C (Configure) in the Action field of the runtime environment you are configuring.
   The Product Component Selection panel displays.
3. Select the component product you want to configure.
   The Configure OMEGAMON XE panel displays.
4. Select Run migration utility. The Run Migration Utility panel will display.
5. Enter the complete Non-VSAM and VSAM high-level qualifiers.
   These values will be used to migrate the following elements:
   - Internal tables database
Running the Migration Utility for IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration (optional)

- MQ configuration database
- Sequential logging dataset

The VSAM copy operation generated by this panel assumes that the new datasets are empty. If they are not empty, the job will receive an error message stating this fact.
Specifying Linkedit Libraries for IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers

Linkedit jobs must be run in the customer environment for the IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers components to run properly. In order to run the linkedit jobs, the Configuration tool must know the names of the appropriate libraries within the system.

Who should read this topic
Read this topic if you are configuring IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers.

Procedure
Perform the following steps to specify linkedit libraries:

1. On the Runtime Environments panel enter C (Configure) in the Action field of the runtime environment you are configuring.
   The Product Component Selection panel displays.
2. Select the component product you want to configure.
   The Configure OMEGAMON XE panel displays.
3. Select Specify link edit libraries.
   The Specify Linkedit Libraries panel will display.
4. Modify the parameters according to the descriptions in the following table, or press Enter to accept the values displayed on each screen that appears for this panel:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MQSeries Authorized load</td>
<td>The library described by the SCSQAUTH low-level qualifier in the IBM MQSeries for MVS/ESA documentation. This field is required.</td>
</tr>
<tr>
<td>MQSeries Load Library</td>
<td>The library described by the SCSQLOAD low-level qualifier in the IBM MQSeries for MVS/ESA documentation. This field is required.</td>
</tr>
<tr>
<td>LE/370 C Dynamic Routines</td>
<td>The library described by the SCEERUN low-level qualifier in the IBM Language Environment documentation. This field is required.</td>
</tr>
<tr>
<td>LE/370 C Static Routines</td>
<td>The library described by the SCEELKED low-level qualifier in the IBM Language Environment documentation. This field is required.</td>
</tr>
<tr>
<td>USS CLIST library</td>
<td>The library described by the SBPXEXEC low-level qualifier in the IBM OS/390 Unix System Service documentation. This field is required.</td>
</tr>
</tbody>
</table>
Creating Configuration Parameters for IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers

These steps are used to display and edit the JCL that creates the configuration parameters for IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers.

Who should read this topic
Read this topic if you are configuring IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers.

Procedure
Perform the following steps to create configuration parameters:

1. On the Runtime Environments panel enter C (Configure) in the Action field of the runtime environment you are configuring.
   The Product Component Selection panel displays.
2. Select OMEGAMON XE for WebSphere Integration Brokers.
   The Configure OMEGAMON XE panel displays.
3. Select Create configuration parameters.
   A panel displays a JCL that creates the configuration parameters for IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers.
4. Review the JCL and submit the job.
   or
   If you want to edit the values in this JCL,
   1. Enter Edit in the command line. The Edit Command Entry panel will display.
   2. Press F1 to access the online help and follow the instructions there.
   3. When the JCL contains the preferred values, press Enter and submit the job.
Completing the Configuration Outside the Configuration Tool
(Common Tasks)

Introduction

This chapter and the following chapter contain topics that correspond to the tasks in the following sections in Chapter 1:

- “Checklist: Configuring IBM Tivoli OMEGAMON XE for WebSphere MQ outside the Configuration Tool” on page 41
- “Checklist: Configuring IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers outside the Configuration Tool” on page 43

Depending on the component products you want to configure, use one or more of these checklists as a guide to configuring the component product outside of the Configuration tool.

Additional, component-product specific, steps are described in “Completing the Configuration Outside the Configuration Tool (Specific to Component Products)” on page 145.

Chapter contents

Copying the Procedures for the Started Task ................................................................. 126
Copying the VTAM Definition and Varying the VTAM Node Active ............................... 127
APF-authorizing Libraries ............................................................................................. 128
Copying the Procedures for the Persistent Data Store .................................................. 129
Installing Product Support for CandleNet Portal and Candle Management Server on a Non-z/OS Machine ................................................................. 130
Seeding the Candle Management Server on a Remote Machine ..................................... 137
Setting up the Configuration Database for Candle Management Server on UNIX .......... 138
Verifying the Configuration: Same Address Space ....................................................... 141
Verifying the Configuration: Different Address Space ................................................ 142
Verifying the Configuration: CMS on a Different Platform ........................................... 143
Copying the Procedures for the Started Task

When you use the Configuration tool to configure a component product, the Configuration tool creates the started task procedures.

You must copy the started task procedures to the procedure library (PROCLIB) to complete the configuration.

Procedure

Copy the started task procedures for the component product from rhileu.midlev RKANSAM (where the variable rhileu.midlev is the runtime dataset high-level qualifier and a middle-level qualifier) to the procedure library (PROCLIB).
Copying the VTAM Definition and Varying the VTAM Node Active

If you configured the component product to use SNA as the communication protocol, the Configuration tool created VTAM definitions.

To complete the configuration, you must do the following:
- Copy the VTAM definition to VTAMLST
- Vary the VTAM major node active

Procedure

Perform the following procedure to complete the configuration for the SNA communication protocol:

1. Copy the VTAM definition from `rhilev.midlev.RKANSAM` (where the variable `rhilev.midlev` is the runtime dataset high-level qualifier and a middle-level qualifier) to VTAMLST.
   
The VTAM definition in `rhilev.midlev.RKANSAM` has the same name you specified using the Configuration tool for either of the following:
   - The VTAM major node
   - The global VTAM major node (The default is KCANDLE1.)

2. Vary the VTAM major node active using the following command:
   ```
   V NET,ACT,ID=ccccccccc
   ```
   
   where the variable `ccccccccc` is the name for the VTAM major node you specified using the Configuration tool for either of the following:
   - The VTAM major node
   - The global VTAM major node
APF-authorizing Libraries

The load libraries must be APF-authorized or some of the functions in the component product may not be available.

You APF-authorize the load libraries by adding them to the list of APF-authorized libraries.

Procedure

Perform one of the following procedures:

- If the runtime environment is sharing with System Modification Program/Extended (SMP/E), APF-authorize the following target load libraries:
  - `thilev.midlev.TKANMOD`
  - `thilev.midlev.TKANMODL`

- For all other runtime environments, APF-authorize the following runtime load libraries:
  - `rhilev.midlev.RKANMOD`
  - `rhilev.midlev.RKANMODL`

where the following are the variables:

  `&thilev.midlev` = the target dataset high-level qualifier and a middle-level qualifier
  `&rhilev.midlev` = the runtime dataset high-level qualifier and a middle-level qualifier

If one library in a steplib or joblib concatenation requires APF-authorization, all the libraries in the concatenation require APF-authorization.

Note: Any runtime libraries concatenated in the STEPLIB DDNAME and in the RKANMODL DDNAME of the started task must be APF-authorized.
Copying the Procedures for the Persistent Data Store

When you configure the Persistent Data Store (CT/PDS), the Configuration tool creates the procedures for the CT/PDS. (See “Configuring the Persistent Data Store” on page 97.) If you decided to configure the CT/PDS, you must copy these procedures to the procedure library (PROCLIB) to complete the configuration.

For more information about the CT/PDS facility, see Historical Data Collection Guide for IBM Tivoli OMEGAMON XE Products.

Procedure

Copy the following procedures from the &shilev.midlev.RKANSAM library (where the variable &shilev.midlev is the SMP/E dataset high-level qualifier and a middle-level qualifier) to the procedure library (PROCLIB):)

- pdsmprefix1
- pdsmprefix2

where the variable pdsmprefix is the unique prefix you specified for the maintenance procedure using the Configuration tool. The default is KPDPROC.
Installing Product Support for CandleNet Portal and Candle Management Server on a Non-z/OS Machine

In order for your product to work properly with OMEGAMON Platform or CandleNet Portal, support for your product must be installed on the same machine as the CandleNet Portal or Candle Management Server (CMS) that will be connecting to it.

On Windows

**Note:** In addition to installing and configuring support for the CMS on a Windows machine, the CMS must be identified at the location of the agent on z/OS in the Configuration tool. See “Changing the CMS connection” on page 114.

Perform the following procedure:

1. Log onto Windows using an ID with Administrator authority and close any running applications.

2. Insert the *IBM Tivoli OMEGAMON XE for WebSphere Business Integration Data Files for z/OS* CD into your CDROM drive. Installation begins automatically. (If the installer does not start, go to CD-ROM directory WINDOWS and run setup.exe. If setup.exe initialization fails, you do not have enough disk space to decompress the setup files.)

   The InstallShield Wizard opens.

3. Read the text that welcomes you to the installation, and click **Next** to continue.

   The Software License Agreement dialog will appear.

4. Read the software license agreement and click **Accept**.

   The Select Features dialog displays.

5. Select the types of support for IBM Tivoli OMEGAMON XE for WebSphere Business Integration that you want to install on this machine.

6. Click **Next**.

   **Note:** The instructions for the rest of this section correspond to installation of all of the support items available on the “*IBM Tivoli OMEGAMON XE for WebSphere Business Integration Data Files for z/OS*” CD when installing it on a Windows machine. If you are installing only particular types of support, or choose to configure only particular types of support, some steps may be unnecessary and the referred-to screens may not appear.

   The Select Program Folder screen will appear.

7. Click **Next** to accept the defaults.

   The Start Copying Files screen will appear. This dialog shows the destination directory and program folder information you supplied in the previous steps, and lists the support you chose to install.

8. Review the settings and click **Back** if you want to go back and change them. If you want to start copying files with the settings that are listed, click **Next**.
Completing the Configuration Outside the Configuration Tool (Common Tasks) 131

Installing Product Support for CandleNet Portal and Candle Management Server on a Non-z/OS Machine

The Setup Status screen will appear and the InstallShield Wizard will begin configuring the installation and copying files needed to complete the installation. Then the Setup Type dialog will appear.

9. Leave selected (checked) the items that you want to configure before completion of the installation. Deselect (uncheck) the items for which you want to delay configuration until after completion of the installation.

Note: If you prefer to seed the CMS at a later time, see “Seeding the Candle Management Server on a Remote Machine” on page 137.

10. Click Next.

The software is copied to disk. Then the Define CNP Host Information dialog displays.

11. Click Next to accept the defaults.

The Candle Management Server Configuration dialog displays.

12. Click OK to accept the defaults.

Depending on whether you configured this CMS to be a remote or a hub CMS when you installed the CMS on this machine, either the Hub CMS Configuration or the Remote CMS Configuration dialog displays.

13. Click OK to accept the defaults.

The WMQ Configurator Data Source Parameters dialog displays.

14. Perform the following procedure to set up configuration database support for IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration:

   1. Select the database type. If you select Internal, you need not specify any additional parameters. If you select ODBC (DB2 UDB), DB2 Universal Database software must already be installed on the machine. See the “Preparing for Installation” chapter of Installing and Setting up OMEGAMON Platform and CandleNet Portal on Windows and UNIX for instructions on how to install the IBM-provided version of DB2.

   2. Complete the rest of the dialog in the following fields:

      A. In the Admin User ID and Admin Password fields, type the user ID (exactly 8 characters) and password of a user with DB2 administrator authority.

      B. In the Database User ID and Database Password fields, either accept the default user ID and password (WMQCFG) or type the user ID and password you want to use for access to the configuration database.

         The user ID must be exactly 8 characters, and the password must be no shorter than 6 characters.

         The data source name for the configuration database is RKCFAPLT. You cannot change this name.

   3. Click OK to save your settings and close the dialog.

Note: If you need to change this initial configuration later, perform the following procedure to get back to the WMQ Configurator Data Source Parameters dialog:

   A. From a command prompt, change to the C:\Candle\INSTALL directory.
B. Enter KCFDataSource.exe.

The Seed CMS dialog is displayed.

15. Select the location of the CMS and click OK.

You will be greeted with a dialog that says that the CMS is not currently running and that, if you choose to continue, it will be started.

16. Click OK.

The Select Product to Seed CMS dialog displays.

The products you chose to install support for are automatically selected on this dialog.

(Seeding adds product-provided situations, templates, and other sample data to the CMS’s Enterprise Information Base (EIB) tables. This can take up to 5 minutes, depending on the number of products you installed.)

**Note:** The Generic Configuration is to seed the cf component of the CMS.

17. Click OK to begin the seeding process.

When seeding is complete, the “Seed data operation complete” dialog displays.

18. Read the log information in the dialog and click Next.

The InstallShield Wizard Complete dialog displays.

19. If you want to view the readme file, make sure “Display the README file” is selected.

20. Click Finish.

The Manage Candle Services window will open, and if previously selected, the readme file will open.

---

**On UNIX (GUI installation)**

**Note:** In addition to installing and configuring support for the CMS on a UNIX machine, the CMS must be identified at the location of the agent on z/OS in the Configuration tool. See “Changing the CMS connection” on page 114.

Perform the following UNIX GUI installation procedure:

**Note:** DO NOT press “Ctrl-C” to stop the installation and configuration process!

The installation and configuration process updates an install database while running. If values requested are not currently available, you should proceed with the process to completion by skipping the entry or entering a dummy value (and noting the location). When the values or resources needed to fill the dummy entries become available later (and before trying to execute the product), you should re-visit the configuration process and fill in the missing values. **If you abort the installation and configuration process pressing “Ctrl-C”, the recommendation is to completely re-install from scratch (for example, using a new $candlehome), as the integrity of the configuration database is no longer assured.**

1. Mount the IBM Tivoli OMEGAMON XE for WebSphere Business Integration Data Files for z/OS CD at the location you have chosen on the host:
Completing the Configuration Outside the Configuration Tool (Common Tasks)

1. 

```
mount device mount_point
cd mount_point
```

where the following are the variables:

- `device` = the device driver for the CD-ROM
- `mount_point` = the directory where the device will be mounted

2. From the bin directory under the root directory of the CD-ROM, execute `install.sh` by entering the following:

```
./install.sh [-h $candlehome] [-d cdrom path] [-v] [-c]
```

where the following are the variables:

Table 46. Parameters for `install.sh` Command

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h</code></td>
<td>An optional parameter used to specify <code>$candlehome</code>. If you do not include <code>-h</code>, and do not have an environmental variable named <code>$candlehome</code> present, <code>install.sh</code> prompts you for the installation directory. Also use this option to take action on a <code>$candlehome</code> other than the <code>$candlehome</code> in the current system.</td>
</tr>
<tr>
<td><code>$candlehome</code></td>
<td>A home directory created for the installation of the CMS</td>
</tr>
<tr>
<td><code>-d</code></td>
<td>An optional parameter used to specify the path to the root of the CD-ROM drive</td>
</tr>
<tr>
<td><code>cdrom path</code></td>
<td>The full path to the root of the CD-ROM drive or the full path to the CD-ROM image</td>
</tr>
<tr>
<td><code>-c</code></td>
<td>An optional parameter used to print diagnostic messages to the console</td>
</tr>
<tr>
<td><code>-v</code></td>
<td>An optional parameter used to display the version and release levels of the current installer</td>
</tr>
</tbody>
</table>

One of the following will occur:

- If the directory specified exists already, you will see a prompt that tells you so and asks you if you want to use this as your home directory.
- If the directory specified does not exist, you will see a prompt that tells you so and asks you if you want to create this as your home directory.

3. Enter `y` and press Enter.

The following prompt to choose the type of installation you want to complete appears.

```
Select one of the following:
1) Install products via GUI.
2) Install products via command line.
3) Create remote packages via GUI.
4) Create remote packages via command line.
5) Exit install.
Please enter a valid number:  1
```

4. Enter the number of the type of install you want to complete. In this case, enter 1.

5. Click **Agree** in the window displaying the license agreement to accept the terms of the license agreement and proceed with the installation.
The Installation screen displays.

6. Click **Install**.

The next screen displays.

7. Click **Install** at the bottom of the screen.

The Install IBM Tivoli OMEGAMON XE dialog displays.

8. Select the IBM Tivoli OMEGAMON XE for WebSphere Business Integration products for which you want to install CMS support data.

9. Click **Install**.

The Manage Candle Services window opens.

10. Perform the following procedure to seed the CMS:

1. Select **Candle Management Server**.

2. Select **Actions > Seed > Advanced Seed**.

   The Seed Products dialog displays.

3. Select the products to seed and click **Seed**.

Seeding for the IBM Tivoli OMEGAMON XE for WebSphere Business Integration products you chose occurs.

**On UNIX (command-line installation)**

*Note:* In addition to installing and configuring support for the CMS on a UNIX machine, the CMS must be identified at the location of the agent on z/OS in the Configuration tool. See “Changing the CMS connection” on page 114.

Perform the following UNIX GUI installation procedure:

*Note:* **DO NOT press “Ctrl-C” to stop the installation and configuration process!**

The installation and configuration process updates an install database while running. If values requested are not currently available, you should proceed with the process to completion by skipping the entry or entering a dummy value (and noting the location). When the values or resources needed to fill the dummy entries become available later (and before trying to execute the product), you should re-visit the configuration process and fill in the missing values. **If you abort the installation and configuration process pressing “Ctrl-C”, the recommendation is to completely re-install from scratch (for example, using a new $candlehome), as the integrity of the configuration database is no longer assured.**

1. Mount the IBM Tivoli OMEGAMON XE for WebSphere Business Integration Data Files for z/OS CD at the location you have chosen on the host:

```
mount device mount_point
cd mount_point
```

where the following are the variables:

- `device` = the device driver for the CD-ROM
- `mount_point` = the directory where the device will be mounted
2. From the bin directory under the root directory of the CD-ROM, execute install.sh by entering the following:

```
./install.sh [-h $candlehome] [-d cdrom path] [-v] [-c]
```

where the following are the variables:

Table 47. Parameters for install.sh Command

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h</td>
<td>An optional parameter used to specify $candlehome. If you do not include -h, and do not have an environmental variable named $candlehome present, install.sh prompts you for the installation directory. Also use this option to take action on a $candlehome other than the $candlehome in the current system.</td>
</tr>
<tr>
<td>$candlehome</td>
<td>A home directory created for the installation of the CMS</td>
</tr>
<tr>
<td>-d</td>
<td>An optional parameter used to specify the path to the root of the CD-ROM drive</td>
</tr>
<tr>
<td>cdrom path</td>
<td>The full path to the root of the CD-ROM drive or the full path to the CD-ROM image</td>
</tr>
<tr>
<td>-c</td>
<td>An optional parameter used to print diagnostic messages to the console</td>
</tr>
<tr>
<td>-v</td>
<td>An optional parameter used to display the version and release levels of the current installer</td>
</tr>
</tbody>
</table>

One of the following will occur:

- If the directory specified exists already, you will see a prompt that tells you so and asks you if you want to use this as your home directory.
- If the directory specified does not exist, you will see a prompt that tells you so and asks you if you want to create this as your home directory.

3. Enter y and press Enter.

The following prompt to choose the type of installation you want to complete appears.

```
Select one of the following:
1) Install products via GUI.
2) Install products via command line.
3) Create remote packages via GUI.
4) Create remote packages via command line.
5) Exit install.

Please enter a valid number: 1
```

4. Enter the number of the type of install you want to complete. In this case, enter 2.

You are given the following prompt:

```
Do you agree to the terms of the license in file LICENSE.TXT on the CD? Enter Yes, No, or R to read the agreement now. (choosing "No" exits the installation process) (Y/N/R):
```

Note: IBM Tivoli has changed its License Agreement which is much longer than the previous agreement. You must read and agree to the license to continue the installation.
5. Enter y to agree with the terms of the license agreement and proceed with the command line installation.

6. Select **CMS support for products running on UNIX** by entering 1.

7. Enter y.

   A list of the products for which you may want to install support will display.

8. Enter the number listed next to your selection.

9. Enter y.

   You will be asked whether you want to install support for those products on a different operating system.

10. Enter n.

11. Perform the following procedure to seed the CMS:

    1. Change to the bin directory under the home directory. Enter the following command:

        \begin{verbatim}
        cd $candlehome/bin
        \end{verbatim}

        where the variable $candlehome is the home directory for the installation.

    2. To start the CMS, enter the following command:

        \begin{verbatim}
        /CandleServer start cms_name
        \end{verbatim}

        where the variable cms_name is the name configured for the CMS.

    3. To seed the CMS, enter the following command:

        \begin{verbatim}
        /CandleSeed -t cms_name pc pc pc pc pc ...
        \end{verbatim}

        where the variables are the following:

        - **cms_name** = the name configured for the CMS
        - **pc** = a two-character product code. For IBM Tivoli OMEGAMON XE for WebSphere Business Integration, they are the following
          - qi for IBM Tivoli OMEGAMON XE for WebSphere Business Integration
          - mq for IBM Tivoli OMEGAMON XE for WebSphere MQ Monitoring
          - mc for IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration
          - cf for the Configurator for IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration

        Seeding for the IBM Tivoli OMEGAMON XE for WebSphere Business Integration products you chose occurs.
Seeding the Candle Management Server on a Remote Machine

You do not need to perform this procedure if, during installation of product support on a non-z/OS Candle Management Server (CMS), you have already seeded the CMS. That procedure is documented in one of the following steps:

- For a CMS on Windows: Step 9. on page 131.
- For a CMS on UNIX via GUI installation: Step 10. on page 134
- For a CMS on UNIX via command-line installation: Step 11. on page 136

When you seed the CMS, you provide the CMS with component product-specific information that the CMS uses for component product-provided solutions (such as predefined situations and policies). To seed the CMS, you will use Manage Candle Services on a remote workstation where you have installed a Candlenet Portal Server.

Procedure

Perform the following steps to seed the CMS using Manage Candle Services:

1. If you have not already done so, install support for IBM Tivoli OMEGAMON XE for WebSphere Business Integration from the IBM Tivoli OMEGAMON XE for WebSphere Business Integration Data Files for z/OS CD on the workstations where the CMS and the Candlenet Portal Server have been installed.

2. On the workstation where you installed the Candlenet Portal Server, select Start > Programs > Candle OMEGAMON XE > Manage Candle Services.

   The program displays the Manage Candle Services window.

3. From the Actions menu in the Manage Candle Services window, select Advanced > Seed CMS....

   The program displays the Seed CMS window.

4. Perform one of the following procedures:

   - For a CMS on your workstation, click the On this computer radio button and click OK.
   - For a CMS installed on a different computer, perform the following procedure:
     A. Click the On a different computer radio button.
        The Non-resident CMS Connection window will display
     B. Type a value in the CMS Node ID field. (Click What’s this? ... for a description of the CMS Node ID field.)
     C. Click the radio button next to the appropriate communications protocol.
     D. Click OK.
        The program displays the Select Product to Seed CMS dialog.

5. Select the component products you want to the seed the CMS for and click OK.

   Manage Candle Services seeds the CMS and displays the Seed Data Operation Complete dialog box indicating whether or not the seeding completed successfully.
What is the configuration database?

When you install IBM Tivoli OMEGAMON XE for WebSphere MQ, you get both a monitoring agent and a configuration agent (IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration). IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration provides a single repository called the configuration database for all your WebSphere MQ configuration data. The configuration database is stored at the hub CMS and includes a default set of objects.

Database types

IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration supports the following two database types:

- the product-provided Internal type, which is installed automatically during CMS installation
- the DB2 Universal Database type, which can be installed from a DB2 Universal Database CD included in the IBM Tivoli OMEGAMON XE for WebSphere Business Integration package. It is recommended that you install the DB2 Universal Database software on the same machine as the hub CMS. See the “Preparing for Installation” chapter of *Installing and Setting up OMEGAMON Platform and CandleNet Portal on Windows and UNIX* for instructions on how to install the IBM-provided version of DB2.

Prerequisites

Make sure the following prerequisites have been satisfied:

- Before you can set up the configuration database, the CMS must be seeded with data for the configuration agent. For instructions, see Step 10. on page 134 or Step 11. on page 136.
- (DB2 Universal Database type only) The DB2 instance owner must create a database and grant permissions to the user who will run the Candle Management Server. Follow the instructions in “(DB2 Universal Database only) Creating a database and granting permissions” on page 139.

Migration

To migrate the configuration database from a previous version, follow these steps.

1. Make sure you have backed up the existing configuration database. For instructions, see the chapter “Backing Up the Configuration Database” in *Using IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration*.
2. Follow the instructions in “Configuration database setup” on page 139.
3. Restore the configuration database. (See the chapter “Backing Up the Configuration Database” in Using IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration.)

**DB2 Universal Database only** Creating a database and granting permissions

If you plan to use the Internal type of database, you can skip this section and go to “Configuration database setup” on page 139.

If you plan to use the DB2 Universal Database, follow these instructions.

1. Log on to DB2 as instance user, where the variable user is the ID of the DB2 instance owner.

2. Start the Command Line Processor for DB2.

3. Execute the following commands to create the database and grant permissions to the user who will run the CMS:

   ```
   CREATE DATABASE RKCFAPLT
   CONNECT TO RKCFAPLT
   GRANT CONNECT ON DATABASE TO USER user
   GRANT CREATETAB ON DATABASE TO USER user
   ```

   where the variable user is the ID of the DB2 instance owner.

4. Follow the instructions immediately below to set up the configuration database.

**Configuration database setup**

After the CMS has been seeded with Configurator data for IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration (For instructions, see Step 10. on page 134 or Step 11. on page 136.), you can set up the configuration database. Perform the following procedure:

1. Stop the CMS by entering the following command:

   ```
   /CandleServer stop cms_name
   ```

   where the variable cms_name is the name configured for the CMS.

2. Execute the CandleConfig script. Enter the following commands:

   ```
   cd /$candlehome/bin
   /CandleConfig -S -t cms_name
   ```

   where the variable $candlehome is the host directory.

   **Note:** For complete information about the CandleConfig script, see “Installing and Setting up OMEGAMON Platform and CandleNet Portal on Windows and UNIX”.

3. At the prompt **User database [Internal or DB2]**, enter the type of database you want to use.

   If you enter **Internal**, you need not specify any additional database parameters.

   If you enter **DB2**, you will be prompted for the name of the DB2 instance and the path to the DB2 home directory.
The data source name for the configuration database is **RKCFAPLT**. You cannot change this name.

4. Restart the CMS. Enter the following command:

   `/CandleServer start < cms_name>`
Verifying the Configuration: Same Address Space

**Note:** The procedure of sharing an agent address space does not apply to the IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers component product. This section applies only to the IBM Tivoli OMEGAMON XE for WebSphere MQ component product.

In addition to completing the configuration for the component product, you must also have completed the configuration of the following:
- The CandleNet Portal interface
- The CMS on the appropriate platform (including seeding the CMS)

**Procedure**

The following illustrates a process to verify the configuration by starting the component product. Undergo this process only if the following is true:
- The CMS is installed on z/OS.
- The component product and the CMS are installed in the same address space.

**Figure 8. Process for Verifying the Configuration: Same Address Space**

1. If the CMS is not already running, vary the CMS VTAM major node active.
2. If the CMS is not already running, start the started task for the CMS.
3. Using the appropriate CandleNet Portal interface, connect to the hub CMS and use the interface to verify that the data for the component product is being displayed.
Verifying the Configuration: Different Address Space

In addition to completing the configuration for the component product, you must also have completed the configuration of the following:

- The CandleNet Portal interface
- The CMS on the appropriate platform (including seeding the CMS)

Procedure

The following illustrates a process to verify the configuration by starting the component product. Undergo this process only if the following is true:

- The CMS is installed on z/OS.
- The agent and CMS are in different address spaces.

**Figure 9. Process for Verifying the Configuration: Different Address Space**

If you selected SNA as the communication protocol for the agent instead of TCP/IP and the CMS is not already running, vary the CMS VTAM major node active. You also need to vary the agent VTAM major node active.

If the CMS is not already running, start the started task for the CMS.

Start the started task for the component product and components you configured.

Using the appropriate CandleNet Portal interface, connect to the hub CMS and use the interface to verify that the data for the component product is being displayed.
Verifying the Configuration: CMS on a Different Platform

In addition to completing the configuration for the component product, you must also have completed the configuration of the following:

- The CandleNet Portal interface
- The CMS on the appropriate platform (including seeding the CMS)

Procedure

Perform this process to verify the configuration by starting the component product. Follow this process only if the following is true:

- You installed the agent on z/OS.
- The CMS is installed on a different platform.

Figure 10. Process for Verifying the Configuration: CMS on a Different Platform

If the CMS is not already running, start CMS on the platform where it is installed. (For information about starting a CMS on a different platform, see the Installing and Setting up guide for that platform.)

Start the started task for the component product and components you configured.

Using the appropriate CandleNet Portal interface, connect to the hub CMS and use the interface to verify that the data for the component product is being displayed.
Completing the Configuration Outside the Configuration Tool (Specific to Component Products)

Introduction

This chapter and the preceding chapter contain topics that correspond to the tasks in the following sections in Chapter 1:

- “Checklist: Configuring IBM Tivoli OMEGAMON XE for WebSphere MQ outside the Configuration Tool” on page 41
- “Checklist: Configuring IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers outside the Configuration Tool” on page 43

Depending on the component products you want to configure, use one or more of these checklists as a guide to configuring the component product outside of the Configuration tool.

Additional, common, steps are described in “Completing the Configuration Outside the Configuration Tool (Common Tasks)” on page 125.

Chapter contents

Granting Authorizations to IBM Tivoli OMEGAMON XE for WebSphere MQ ................. 146
Enabling Monitoring of CICS Channels for IBM Tivoli OMEGAMON XE for WebSphere MQ ................................................................. 149
Multi-level Security: Reviewing Configuration Database Security in IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration ........................................ 150
Multi-level Security: Implementing Configuration Database Security in IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration .......................... 159
Authorizing IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers ........ 161
Installing the CandleMonitor Node in Broker Environments .............................. 163
Setting Parameters for IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers ........ 165
Granting Authorizations to IBM Tivoli OMEGAMON XE for WebSphere MQ

Who should read this topic
Read this topic if you are configuring IBM Tivoli OMEGAMON XE for WebSphere MQ.

Procedures
This section provides guidelines for authorizing the external security program (for example, Resource Access Control Facility (RACF)) to permit the following:

- IBM Tivoli OMEGAMON XE for WebSphere MQ Monitoring to access the queue managers
- IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration to configure the queue managers

Authorization can be granted to agents that run either in the Candle Management Server (CMS) address space or in their own address space. The following table lists the authorizations required by component products:

Table 48. Authorizations for Agents

<table>
<thead>
<tr>
<th>Agent</th>
<th>Required Authorizations</th>
</tr>
</thead>
</table>
| IBM Tivoli OMEGAMON XE for WebSphere MQ Monitoring | - Read or browse WebSphere MQ event queues. The agent or CMS identifier must also be granted authority to alter the QDPMAXEV parameter on monitored queues.  
- Get messages from and put messages to monitored queues.  
- Browse the WebSphere MQ queue SYSTEM.CHANNEL.SEQNO for CICS channel support. |
| IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration | - Connect to the WebSphere MQ queue manager (MQCONN).  
- Issue system commands to WebSphere MQ and receive responses.  
- Get messages from and put messages to IBM Tivoli queues starting with KMC (the default) or the high-level qualifier for the site (hilev). |

How you set these authorizations depends upon the external security program and the parameters that you specify for IBM Tivoli OMEGAMON XE for WebSphere MQ Monitoring. Refer to the RACF and CA-TOP SECRET examples below. Also, refer to Using IBM Tivoli OMEGAMON XE for WebSphere MQ Monitoring and the documentation for the external security program.

Resource Access Control Facility example
This is an example of authorization procedures using RACF. If you use another security manager, see the required authorizations listed above.
1. For each subsystem that you plan to monitor, issue the following WebSphere MQ command:

   **DISPLAY SECURITY SWITCHES**

   The current security switch profiles settings for the specified subsystem display.

2. Perform one of the following procedures:

   - If the following message displays, then you do not need to grant RACF authorizations for the specified subsystem. Skip the remaining steps in this procedure.

     **CSQH001I CSQHPDTC SUBSYSTEM SECURITY SWITCH SET OFF,**

   - If the above message does not display, proceed to the next step in this procedure.

3. If a RESLEVEL profile has not been defined in RACF class MQADMIN for this WebSphere MQ subsystem, then define one by issuing the following RACF command:

   **RDEFINE MQADMIN ssid.RESLEVEL UACC(NONE) AUDIT(ALL),**

   where the variable `ssid` is the WebSphere MQ subsystem.

   You have defined the RESLEVEL profile in RACF security class MQADMIN for WebSphere MQ subsystem `ssid`.

4. Issue the following RACF command:

   **PERMIT ssid.RESLEVEL CLASS(MQADMIN) ID(auser) ACCESS(CONTROL)**

   where the variable `auser` is the userid for the agent’s started task.

   You have granted the userid for the agent’s started task access to all WebSphere MQ resources.

5. Issue the following RACF command:

   **RDEFINE MQCMDS ssid.** UACC(NONE)

   You have defined the profile in RACF security class MQCMDS.

6. Issue the following RACF command:

   **PERMIT ssid.DISPLAY.** CLASS(MQCMDS) ID(auser) ACCESS(READ)

   You have granted the userid for the agent’s started task the ability to issue all WebSphere MQ commands.

**CA-TOP SECRET® example**

This is an example of authorization using CA-TOP SECRET security. It is intended as a guide only; you will need to adapt these notes to conform to the release and configuration of the security product. See the required authorizations listed above.

1. If necessary, make the MQM facility resident. Enter the following:

   **TSS MODIFY(FAC(MQM=RES))**

2. Provide access to WebSphere MQ facilities for the ACID (the userid for the agent’s started task). Enter the following:
TSS ADD(auser) MQADMIN(csq1.)
TSS ADD(auser) MQCMDS(csq1.)
TSS ADD(auser) MQQUEUE(csq1.)
TSS PER(auser) MQADMIN(csq1.RESLEVEL)
TSS PER(auser) MQCMDS(csq1.*) ACCESS(ALL)
TSS PER(auser) MQQUEUE(csq1.*) ACCESS(ALL)

where the following are the variables:

- \texttt{auser} = the ACID (the userid for the agent’s started task)
- \texttt{csq1} = the queue manager name

3. Refresh the WebSphere MQ security for the queue manager by issuing the following command from z/OS:

\texttt{xxxx REVERIFY SECURITY(auser)}

where the following are the variables:

- \texttt{xxxx} = the command prefix for the \texttt{csq1} queue manager
- \texttt{auser} = the ACID (the userid for the agent’s started task)
Enabling Monitoring of CICS Channels for IBM Tivoli OMEGAMON XE for WebSphere MQ

Who should read this topic
Read this topic if you are configuring IBM Tivoli OMEGAMON XE for WebSphere MQ.

Procedure
Perform the following steps to enable monitoring of CICS channels for IBM Tivoli OMEGAMON XE for WebSphere MQ:

1. Edit the CICS region Job Control Language (JCL). Do the following:
   1. Locate the DFHRPL DD statement.
   2. Add the KMQAMQCI library to the concatenation. The following is an example:
      ```
      CANDLE.CCCMQMC.TKANMOD(KMQAMQCI)
      ```
2. Using the online resource definition (transaction CEDA), define program KMQAMQCI. The language is Assembler and EXECKEY is CICS.
3. Using CEDA, define a transaction that executes program KMQAMQCI. TASKDATAKEY is CICS. IBM recommends using transaction ID KMQC.
4. The default high-level qualifier for the IBM-Tivoli-created queues is **KMQ**. If you defined a different high-level qualifier, you must specify it using INITPARM in the CICS System Initialization Table (SIT) or in the SYSIN override. Enter the following:
   ```ini
   INITPARM=(KMQAMQCI='HLQ=hilev')
   ```
   where the variable *hilev* is the new high-level qualifier that you defined.
5. Start transaction KMQC using one of the following methods:
   - Add program KMQAMQCI to the PLTPI. Use INITPARM if you want to specify a high-level qualifier.
   - Enter transaction KMQC START [HLQ=\textit{hilev}] at a CICS terminal.
   - Link to program KMQAMQCI, passing START [HLQ=\textit{hilev}] in the common area.
   - Issue an EXEC CICS START command for transaction KMQC, passing START [HLQ=\textit{hilev}] as data.
6. If you wish to stop KMQC manually, (Transaction KMQC is automatically stopped at CICS shutdown.) use one of the following methods:
   - At a CICS terminal, enter the following command:
     ```
     KMQC STOP
     ```
   - Link to program KMQAMQCI, passing STOP in the common area.
   - Issue an EXEC CICS START command for transaction KMQC, passing STOP as data.
Multi-level Security: Reviewing Configuration Database Security in IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration

Who should read this topic
Read this topic if you are configuring IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration.

Multi-level security feature
The multi-level security feature for the z/OS-based configuration database resides with the hub CMS.

Using the multi-level security feature you can restrict access to the configuration database at up to four increasingly resource-specific levels. Known collectively as “authorization-checking levels”, they correspond to the following object classifications in the environment for IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration:

- Configured System Group
- Configured System
- Resource Group
- Resource

Implementation of this security feature involves setting one of these four authorization-checking levels for each Configured System Group that you define. It also involves the creation of RACF profiles that are associated, by naming convention, with the level you have set.

When you attempt to manipulate an object (resource) within a protected Configured System Group, the z/OS System Authorization Facility (SAF) searches for the profile associated with the assigned authorization-checking level and uses its access attributes.
Multi-level Security: Reviewing Configuration Database Security in IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration

Authorization-checking levels

The following is a description of the four authorization-checking levels, including the hierarchy of resources covered by each level:

<table>
<thead>
<tr>
<th>Authorization-checking Level</th>
<th>Hierarchy Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configured System Group</td>
<td>Top</td>
<td>Implies that only one profile exists for each Configured System Group (group of queue managers) and that all resources within the group are covered by that profile. When Configured System Group is specified as the authorization-checking level, SAF begins searching for that profile only, even if other profiles associated with lower authorization-checking levels exist. If it does not find the designated profile, it grants access. This is the level with the least system overhead; however, it affords the least flexibility in restricting access to objects (resources).</td>
</tr>
<tr>
<td>Configured System</td>
<td>Second</td>
<td>(The default setting.) Implies that one profile exists for each Configured System (queue manager) and that all resources associated with that Configured System are covered by that profile. When Configured System is specified as the authorization-checking level, SAF searches for that profile only, even if other profiles associated with lower authorization-checking levels exist. If it does not find the designated profile, it searches for a profile associated with the parent Configured System Group. If it does not find that profile either, it grants access.</td>
</tr>
<tr>
<td>Resource Group</td>
<td>Third</td>
<td>Implies that one profile exists for each Resource Group (group of queues, channels, namelists, etc.) and that all resources associated with that Resource Group are covered by that profile. When Resource Group is specified as the authorization-checking level, SAF begins searching for that profile, even if other profiles associated with the lowest authorization-checking level (Resource) exist. If it does not find the designated profile, it searches for a profile associated with the parent Configured System. If it does not find that profile either, it grants access.</td>
</tr>
</tbody>
</table>
Guidelines for creating Resource Access Control Facility profiles

This section contains rules and guidelines to help you create the necessary profiles to protect the objects in the configuration database.

When you create a profile, you will assign user IDs certain access levels to it. The following access levels define each user’s control over the objects protected by that profile:

Table 50. Resource Access Control Facility Profile User Access Levels

<table>
<thead>
<tr>
<th>Access Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NONE</td>
<td>The user cannot even view the object’s settings list.</td>
</tr>
<tr>
<td>READ</td>
<td>The user can view the object but not change it.</td>
</tr>
<tr>
<td>UPDATE</td>
<td>The user can make changes to the object.</td>
</tr>
<tr>
<td>CONTROL</td>
<td>The user can make changes to the object and perform Update Actual from Defined against the object.</td>
</tr>
<tr>
<td>ALTER</td>
<td>The user has full access to the object, including the ability to define or delete it.</td>
</tr>
</tbody>
</table>

For example, assume you want to create a profile that gives full authority to user MQADMIN for a queue manager named QMGRPROD running on system SYSPROD. All other users will have READ access only. The following commands would create that profile:

```
RDEFINE $KCFADMN MQ.QMGR.SYSPROD.QMGRPROD UACC(READ)
PERMIT QMGR.SYSPROD.QMGRPROD CLASS($KCFADMN) ID(MQADMIN) ACCESS(ALTER)
```

The general types of profiles that you can create are those that cover the following:

- Objects in defined configurations
- Objects in prototype configurations
- Special tasks

Table 49. Authorization-Checking Levels (continued)

<table>
<thead>
<tr>
<th>Authorization-checking Level</th>
<th>Hierarchy Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource</td>
<td>Bottom</td>
<td>Implies that one profile exists for each individual Resource. When Resource is specified as the authorization-checking level, SAF begins searching for that profile. If it does not find the designated profile, it searches for a profile associated with the parent Resource Group. If it does not find that profile either, it searches for a profile associated with the parent Configured System, and then the parent Configured System Group. If it does not find either of those profiles, it grants access.</td>
</tr>
</tbody>
</table>
For any resource(s) that possess common access attributes, you can also create generic profiles.

All these profile types are described in the following sections. While reading these sections, recall that RACF profile names do not support lower-case characters. When an authorization-check is performed, lower-case characters are converted to uppercase. So, assuming that on SYS1 there were two queue managers, one called QMGR1 and one called qmgr1, both would be protected by the same RACF profile called MQ.QMGR.SYS1.QMGR1.

**Defined configurations**

In defined configurations, you associate the authorization-checking level with the profile by adhering to profile-naming conventions. These naming conventions are described in the following table.

In this table, required characters in each profile name are represented as capital letters (for example, MQ.QMGR). The remainder of the string, represented by lower-case, italicized characters (for example, system-name.queue-manager-name) is the site-specific value.

**Table 51. Profile Naming Conventions for Defined Configuration Objects**

<table>
<thead>
<tr>
<th>Level</th>
<th>Profile Naming Convention</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configured System</td>
<td>CSG.configured-system-group-name</td>
<td>The variable configured-system-group-name is the name of the particular configured system group covered by this profile.</td>
</tr>
<tr>
<td>Group</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Configured System</td>
<td>MQ.QMGR.system-name.queue-manager-name</td>
<td>▪ The variable system-name is the name of the host system where this configured system (queue manager) resides.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>▪ The variable queue-manager-name is the name of the particular configured system (queue manager) covered by this profile.</td>
</tr>
</tbody>
</table>
Table 51. Profile Naming Conventions for Defined Configuration Objects (continued)

<table>
<thead>
<tr>
<th>Level</th>
<th>Profile Naming Convention</th>
<th>Explanation</th>
</tr>
</thead>
</table>
| Resource Group | parent-object-profile-name.RG.resource-group-name | - The variable parent-object-profile-name is the full name of the profile for the parent of this resource group.  
- The variable resource-group-name is the name of the resource group covered by this profile.  
For example, assume there is a resource group named “myresource”, existing within a queue manager named “myqmgr” on “systema”. The profile that protects this resource would be named: MQ.QMGR.systema.myqmgr.RG.myresource |
| queue          | MQ.QMGR.system-name.queue-manager-name.QUEUE.queue-name | - The variable system-name is the name of the host system where the queue manager resides.  
- The variable queue-manager-name is the name of the queue manager of this queue.  
- The variable queue-name is the name of the queue covered by this profile. |
| channel        | MQ.QMGR.system-name.queue-manager-name.CHANNEL.channel-name | - The variable system-name is the name of the host system where the queue manager resides.  
- The variable queue-manager-name is the name of the queue manager of this queue.  
- The variable channel-name is the name of the channel covered by this profile. |
### Table 51. Profile Naming Conventions for Defined Configuration Objects (continued)

<table>
<thead>
<tr>
<th>Level</th>
<th>Profile Naming Convention</th>
<th>Explanation</th>
</tr>
</thead>
</table>
| process           | MQ.QMGR.system-name. queue-manager-name. PROCESS.process-name                           | - The variable system-name is the name of the host system where the queue manager resides.  
- The variable queue-manager-name is the name of the queue manager of this queue.  
- The variable process-name is the name of the process covered by this profile. |
| namelist          | MQ.QMGR.system-name. queue-manager-name. NAMELIST.namelist-name                          | - The variable system-name is the name of the host system where the queue manager resides.  
- The variable queue-manager-name is the name of the queue manager of this queue.  
- The variable namelist-name is the name of the namelist covered by this profile. |
| storage class     | MQ.QMGR.system-name. queue-manager-name. STGCLASS.storage-class-name                     | - The variable system-name is the name of the host system where the queue manager resides.  
- The variable queue-manager-name is the name of the queue manager of this queue.  
- The variable storage-class-name is the name of the storage class covered by this profile. |
| authentication information | MQ.QMGR.system-name. queue-manager-name. AUTHINFO.authinfo-name | - The variable system-name is the name of the host system where the queue manager resides.  
- The variable queue-manager-name is the name of the queue manager of this queue.  
- The variable authinfo-name is the name of the authentication information covered by this profile. |
Prototype configurations

In prototype configurations, there is no relationship between the authorization-checking level and the profile name; SAF always searches for a prototype-specific profile. Naming conventions for prototype profiles are described in the following table.

Required characters in each profile name are represented as capital letters (for example, MQ.PROTO.QMGR). The remainder of the string, represented by lower-case, italicized characters (prototype-name), is the prototype name.

Table 51. Profile Naming Conventions for Defined Configuration Objects (continued)

<table>
<thead>
<tr>
<th>Level</th>
<th>Profile Naming Convention</th>
<th>Explanation</th>
</tr>
</thead>
</table>
| coupling facility          | MQ.QMGR.system-name. queue-manager-name. CFSTRUCT.cfstruct-name                          | - The variable system-name is the name of the host system where the queue manager resides.  
- The variable queue-manager-name is the name of the queue manager of this queue.  
- The variable cfstruct-name is the name of the coupling facility covered by this profile. |

Table 52. Profile Naming Conventions for Prototype Configuration Objects

<table>
<thead>
<tr>
<th>Prototype</th>
<th>Profile Naming Convention</th>
<th>Explanation</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>resource group</td>
<td>PROTO.RG.resource-group-name</td>
<td></td>
<td>The variable resource-group-name is the name of the resource group prototype.</td>
</tr>
<tr>
<td>queue manager</td>
<td>MQ.PROTO.QMGR. prototype-name</td>
<td></td>
<td>The variable prototype-name is the name of the queue manager prototype.</td>
</tr>
<tr>
<td>queue</td>
<td>MQ.PROTO.QUEUE. prototype-name</td>
<td></td>
<td>The variable prototype-name is the name of the queue prototype.</td>
</tr>
<tr>
<td>channel</td>
<td>MQ.PROTO.CHANNEL. prototype-name</td>
<td></td>
<td>The variable prototype-name is the name of the channel prototype.</td>
</tr>
<tr>
<td>process</td>
<td>MQ.PROTO.PROCESS. prototype-name</td>
<td></td>
<td>The variable prototype-name is the name of the process prototype.</td>
</tr>
<tr>
<td>namelist</td>
<td>MQ.PROTO.NAMELIST. prototype-name</td>
<td></td>
<td>The variable prototype-name is the name of the namelist prototype.</td>
</tr>
<tr>
<td>storage class</td>
<td>MQ.PROTO.STGCLASS. prototype-name</td>
<td></td>
<td>The variable prototype-name is the name of the storage class prototype.</td>
</tr>
</tbody>
</table>
Tasks

In addition to creating profiles to restrict access to objects in both defined and prototype configurations, you can create profiles to restrict users from performing certain tasks.

RACF profiles can be created to restrict users from the following:

- Creating, modifying, or deleting scheduled actions
- Editing global variables
- Viewing and manipulating the enhanced audit log

These task-oriented profiles are described in the following table. In this table, required characters in each profile name are represented as capital letters (for example, ADMIN.SCHEDUPT). The remainder of the string, represented by lower-case, italicized characters (for example, scheduled-update-name) is the site-specific value.

<table>
<thead>
<tr>
<th>Task</th>
<th>Profile Naming Convention</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scheduled actions</td>
<td>ADMIN.SCHEDUPT.</td>
<td>The variable scheduled-update-name is the name of the scheduled update.</td>
</tr>
<tr>
<td></td>
<td>scheduled-update-name</td>
<td>The creator of a scheduled update always has access to it.</td>
</tr>
<tr>
<td>Editing of global variables</td>
<td>ADMIN.GLVAR</td>
<td>Users must have UPDATE access to alter, create, or delete global variables.</td>
</tr>
<tr>
<td>Manipulation of enhanced audit log</td>
<td>ADMIN.AUDIT</td>
<td>Users must have at least READ access to view the audit log.</td>
</tr>
</tbody>
</table>

Generic profiles

Where access attributes are the same across resources (or tasks), you can save time by creating generic profiles. Generic profiles are created by using standard RACF wildcards (*, **, %). (Refer to the RACF documentation for more information on wildcards.)

Note: SAF searches for the most specific applicable profile. If you create both a resource-specific profile and a generic profile that covers that resource, the specific profile is used.

For example, assume that the authorization-checking level for a certain Configured System Group is Resource and the following two profiles exist:

- MQ.QMGR.SYSA.QMGR1.QUEUE.QUEUE1
  This profile restricts access to a queue named QUEUE1 on a queue manager named QMGR1 on a system named SYSA.
- MQ.QMGR.SYSA.*.QUEUE.*
  This profile restricts access to all queues on all queue managers on SYSA.

If you attempt to access QUEUE1, SAF uses the access attributes in the first profile (MQ.QMGR.SYSA.QMGR1.QUEUE.QUEUE1) to determine your authority.
Examples of generic profiles
Following are more examples of generic profiles. Refer to the tables above, if necessary, to review required characters in the naming conventions (remember that lower-case characters are converted to uppercase):

- **ADMIN.***
  This profile does the following:
  - Restricts creation, modification, and deletion of all scheduled updates
  - Restricts editing of global variables
  - Restricts viewing and manipulation of the enhanced audit log.

- **ADMIN.SCHEDUPT.***
  This profile restricts creation, modification, and deletion of all scheduled updates.

- **MQ.QMGR.SYSTEM*.QMG*.PROCESS.myprocess**
  This profile restricts access to all processes named “myprocess” on any queue manager that begins with “qmg” on any system that begins with “system”.

- **CSG.***
  This profile restricts access to all configured system groups.

- **Proto.Rg.***
  This profile restricts access to all resource groups.
Multi-level Security: Implementing Configuration Database Security in IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration

Who should read this topic

Read this topic if you are configuring IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration.

Before implementing configuration database security

Before you begin to implement configuration database multi-level security, you must create a new RACF class, which requires that you re-IPL z/OS.

Implementing security from the z/OS terminal

Perform the following steps on the z/OS system to implement security:

1. Inspect member KDSENV in the RKANPAR dataset (for example, &rhilev.RKANPAR(KDSENV), where the variable &rhilev is the runtime dataset high-level qualifier), locate the following environment variable defaults, and verify that the settings for each are as listed in the following:

   KDS_KCF_EXT_SECURITY=YES
   KDS_KCF_SAFCLASS=$KCFADMN
   KDS_VALIDATE=YES

   If you modify any of these values, you must add them to the end of the KDSENV member in the format shown above. These statements, respectively do the following:

   ■ Enable this component product’s multi-level security feature
   ■ Set the default name of a new RACF class for the configuration database profiles (you may change the default name of $KCFADMN)
   ■ Enable user ID checking at the CMS (where the configuration database resides)

2. Define the new $KCFADMN class. If necessary, refer to the sample source in member KCFRRCDE in TKANSAM.

3. Assemble and link the ICHRRCDE table.

4. Enter the new class in the RACF router table (ICHRFR01). Source for this entry can be found in TKANSAM, member KCFRFR01.

5. Re-IPL z/OS.

6. Have the Security Administrator (or someone else with RACF SPECIAL authority) issue the following command to activate the new RACF class:

   SETROPTS CLASSACT($KCFADMN) GENERIC($KCFADMN) RACLIST($KCFADMN)

7. From a RACF session, create profiles within $KCFADMN, using the rules and guidelines in the above sections.
Implementing security from CandleNet Portal

Perform the following steps in CandleNet Portal to implement security:

1. Log on to IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration. (From the list of available navigator views, select the Configuration view.)

2. Ensure you are in update mode.

3. Open the Defined View.

4. Select the Configured System Group that you wish to assign an authorization-checking level to (or define a new Configured System Group and select it).

5. In the Group section of the settings list, locate the Authorization Checking Level parameter. Select the level that you wish to assign to this Configured System Group, or select None.

   When a user attempts to access any resource in that Configured System Group, SAF will begin searching for a profile that corresponds (by naming convention) to that level. If none is found, it will search at the next higher level.

6. Select Save to save the changes.
Authorizing IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers

This section describes the actions required in order to authorize IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers to receive the correct broker data.

Who should read this topic
Read this topic if you are configuring IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers.

Procedure
To authorize IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers on z/OS, perform the following steps:

1. If you have not already done so, follow the procedure in “APF-authorizing Libraries” on page 128.
   (RKANMOD and TKANMODL (or, depending on the runtime environment configuration, RKANMODL) datasets for the agent's started task procedure need to be APF authorized.)

2. Ensure the userid for the agent's started task has read access to files in the monitored brokers' component directories.

3. Ensure the userid for the agent's started task has the following:
   - Proper authorization to MQOPEN and MQPUT to SYSTEM.BROKER.ADMIN.QUEUE and SYSTEM.BROKER.CONTROL.QUEUE
   - Proper authorization to use SYSTEM.BROKER.MODEL.QUEUE as a model queue during MQOPEN processing
   - Proper authorization to MQOPEN (for create) and MQGET from KQI.AGENT.REPLY.QUEUE (or the name the site defined in agent parameters to be the agent reply queue)

4. Ensure the userid for the agent’s started task procedure is specifically permitted (in RACF) to the UNIXPRIV class SUPERUSER.PROCESS.GETPSEND. (See the instructions below.)

**Note:** Otherwise, these symptoms will occur: message KQIA153E giving return code 319030247 in the agent log; missing job name, start date and time, and ASID information for a broker or execution group; and sporadic reporting that a broker or execution group has a status of "started" when it is not.

UNIXPRIV SUPERUSER.PROCESS.GETPSEND is described in the section “Using UNIXPRIV Class Profiles” in the chapter "Establishing UNIX Security" of the IBM manual UNIX System Services Planning. The following are instructions required to set this up for the agent:

1. Define a profile in the UNIXPRIV class to protect the resource called SUPERUSER.PROCESS.GETPSEND. Enter the following:
   ```
   RDEFINE UNIXPRIV SUPERUSER.PROCESS.GETPSEND UACC(NONE)
   ```
2. Assign the agent's started task procedure userid (xxxxxxx). Enter the following:
   
   ```
   PERMIT SUPERUSER.PROCESS.GETPSENG CLASS(UNIXPRIV) ID(xxxxxxx)
   ACCESS(READ)
   ```

3. Activate the UNIXPRIV class, if it is not currently active at the installation. Enter the following:
   
   ```
   SETROPTS CLASSACT(UNIXPRIV)
   ```

4. Activate SETROPTS RACLST processing for the UNIXPRIV class, if it is not already active. Enter the following:
   
   ```
   SETROPTS RACLST(UNIXPRIV)
   ```

5. If SETROPTS RACLST processing is already in effect for the UNIXPRIV class, refresh SETROPTS RACLST processing so that the changed profile in the UNIXPRIV class will take effect. Enter the following:
   
   ```
   SETROPTS RACLST(UNIXPRIV) REFRESH
   ```

**Additional authorization task**

Additionally, depending on the site's ACL entries, you may be required to authorize the agent to receive broker event publications in the WebSphere Integration Brokers Control Center. You do not have to perform this step if the site uses the IBM defaults. However, if ACL entries in the Topics tab of the WebSphere Integration Brokers Control Center have been modified such that subscribe access to topics beginning with $SYS/Broker has been restricted, this step is required.

An ACL entry for topics beginning with $SYS/Broker must be added to set Subscribe access to allow for the userid of the agent. The principal for the ACL entry should give the userid of the agent, or it can give a group to which the agent's userid belongs, such as “mqbrkrs”. The ACL entry must be deployed to all brokers to be monitored by IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers. This change will allow the agent to receive the broker event publications. If the agent is restricted from receiving these publications, much data in IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers reports will be missing or inaccurate. (Specifically, the Broker Events workspace will display no data.)
Installing the CandleMonitor Node in Broker Environments

The CandleMonitor Node is an optional component that collects statistics on message flow performance in a broker. It also provides a mechanism for generating user-defined events within a message flow.

With V5 brokers, IBM now provides message flow accounting and statistics that partially overlap with the statistics provided by the CandleMonitor node and can be monitored in CandleNet Portal without including the CandleMonitor node in message flows. However, the CandleMonitor node provides additional statistics, as well as the ability to monitor sub-flows and define message flow events.

**Note:** Information about configuring message flows with the CandleMonitor node can be found in “Using IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers” in the chapter “Customizing the CandleMonitor node.”

If you do not plan to use the CandleMonitor Node in message flows, you can skip the rest of this section.

Who should read this topic

Read this topic if you are configuring IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers.

Procedure

Before the CandleMonitor node can be used to monitor message flows, it must be installed in the broker environments. In addition it must be made available in the Control Center (V2.1) or Message Brokers Toolkit (V5 or V5.1), as documented in Using IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers in the chapter “Customizing the CandleMonitor node.”

On z/OS, the CandleMonitor node can be installed either permanently or temporarily.

**Note:** During installation and configuration, the HFS Candlehome directory is created with a user-specified name. That directory is identified as <Candlehome> in the statements below. The broker's user ID must have read access to all files and subdirectories in the <Candlehome>/kqi directory.

Permanent installation

To permanently install the CandleMonitor node in a broker environment on z/OS, do the following:

1. In the broker’s component directory, find and edit the mqsicompcri file for the broker. Make the following changes:
   - Change the LILPATH= statement to add the IBM Tivoli path by appending to the end of the existing statement, as in the following:

     \:<Candlehome>/kqi/lil
- Change the environment variable section to add the KQI_CONFIG_FILE_PATH variable by adding the following line after (ENVIRONMENTBEGIN) and before (ENVIRONMENTEND):

  KQI_CONFIG_FILE_PATH=<Candlehome>/kqi/lil

2. Save and exit the mqsicompcif file.

3. From the same broker component directory, issue the mqsicustomize command. Note that none of the changes made to the mqsicompcif file will require that you re-run any other customization job or command.

4. Issue the following command in USS with a user ID with proper authority. Replace <Candlehome> as indicated at the beginning of this section, and replace <InstallationPath> with the broker's installation path.


5. Restart the broker.

**Temporary installation**

Temporary installation of the CandleMonitor node will modify the broker's environment only until the next time mqsicustomize is run, unless steps are taken to make the change permanent, as described above.

To temporarily install the CandleMonitor node in a broker environment on z/OS, do the following:

1. In the broker’s HFS component directory, find and edit the ENVFILE file for the broker. Add the following statement to the end of the file:

   KQI_CONFIG_FILE_PATH=<Candlehome>/kqi/lil

2. Save and exit the ENVFILE.

3. Issue the following command in USS with a user ID with proper authority. Replace <Candlehome> as indicated at the beginning of this section, and replace <InstallationPath> with the broker's installation path.


4. Issue the following commands, in order, on the z/OS operator console. Note that these commands restart the broker component.

   F stpn,PC
   F stpn,CB L='existing lil path>:<Candlehome>/kqi/lil'
   F stpn,SC

where the following are the variables:

- stpn = the broker's started task procedure name
- existing lil path = the current LILPATH defined in the mqsicompcif file
Setting Parameters for IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers

Who should read this topic

Read this topic if you are configuring IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers.

Where to find instructions for this procedure

You can change the parameters that determine the operational and monitoring characteristics of IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers. The parameters reside in a file in XML format that is created during the installation of IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers.

See the chapter "Parameters and Performance Considerations" in Using IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers for information about setting the agent parameters.
9

Updating and Changing the Configuration

Introduction

This chapter contains topics that correspond to the tasks in the section in Chapter 1 called “Checklist: Updating and Changing the Configuration” on page 45.

Chapter contents

Updating and Changing the Configuration ......................................................... 168
**Example of a change to the configuration that requires manual steps**

If the changes you make in the Configuration tool are to configuration values that also require manual steps outside of the Configuration tool, you must also perform the steps outside of the Configuration tool when you update or change the values in the Configuration tool.

For example, if you are using the Persistent Data Store (CT/PDS) and you change the prefix for the procedure using the Configuration tool, you must also copy the renamed procedure to the PROCLIB.

For step-by-step instructions for these tasks, see “Completing the Configuration Outside the Configuration Tool (Common Tasks)” on page 125.

**Procedure**

The following illustration shows the process for updating or changing the configuration:

**Figure 11. Process for Updating and Changing the Configuration**

1. Using the Configuration tool, make the changes you want to the configuration.
2. On the Runtime Environments panel in the Configuration tool, use the L (Load libraries after SMP/E) select to load the target libraries to the runtime libraries.
3. If the changes are to values that require manual steps outside of the Configuration tool to complete the configuration, perform the appropriate steps.
Enabling System Variable Support

Introduction

This chapter provides detailed instructions for the following:

- Enabling system variable support
- Using system variable support to run the component products on any z/OS system

Chapter contents

Background on System Variable Support .................................................. 170
Sample Usage Scenario ................................................................. 171
Enabling System Variable Support .................................................... 173
Creating the System Variable Parameter Member ................................. 175
Creating the VTAM Major Node Rename Job .................................... 176
Creating a VTAM Major Node (One Node for All Component Products) .... 177
Background on System Variable Support

Utilizing system variable support

System variable support provides you with an easy method of running software on any of the z/OS systems. The software is configured so that it becomes z/OS system-independent. It can then be ported and started on any z/OS system without extensive Configuration tool reconfiguration.

Note: For additional information, you can access the README file for system variable support by issuing the README SYS command.

By utilizing system variable support, the component products inherit the system values for the system on which they are started (the host z/OS system). These system-specific values are then automatically loaded into dynamic in-memory parameter members that exist only for the duration of the component product execution. The result is that the software runs correctly using the system-specific parameter values for the host z/OS system.

Benefits of using system variable support

The following are some of the benefits of using system variable support:

- You can deploy the same software unit, consisting of any or all component products, on any system without modification. LPAR-specific values are automatically resolved and substituted at component product startup.
- The number of unique runtime environments required is less, which can translate into savings on DASD, CPU, and human time.
- The same started task Job Control Language (JCL) and the same VTAM node can be used on any system without modification.
- You can choose to use a single VTAM major node in place of the individual component product major nodes. When generated, it will contain all VTAM applids for all component products you have configured in the runtime environment.

Considerations when using system variable support

When using system variable support, you should consider the following:

- Component product started tasks (STCs) contain a new preprocessing step (STEP1 in the STC JCL). This new step resolves all system variable specifications in the component product parameter members.
- Component product parameter members contain many variables, instead of values, that are resolved when the STC starts. For example, the members contain &SVXDSNV instead of the VSAM high-level qualifier value.
Sample Usage Scenario

Process to enable and use system variable support

The following steps describe the process for enabling and using system variable support:

1. Define and configure a runtime environment. During runtime environment configuration, specify the values to enable system variable support. See “Setting up the Runtime Environment” on page 79 for details on configuring a runtime environment.

   **Note:** *Multiple runtime environments may be required depending on the runtime environment type (FULL, SHARING, BASE, or SMP/E), the Candle Management Server (CMS) type (hub or remote), and variations in component product mixtures.*

2. Create the system variable parameter member.

   After configuring a runtime environment, you can create the system variable parameter member. Each runtime environment contains one user system variable parameter member named RKANPAR(midlev), which contains all system-specific values.

   See “Creating the System Variable Parameter Member” on page 175 for information on creating the midlev system variable parameter member in the RKANPAR library.

3. Create the VTAM major node rename job.

   After configuring a runtime environment, you can create the VTAM major node rename job. This job creates VTAM major nodes on remote systems with names that are resolved from the system variable specification.

   See “Creating the VTAM Major Node Rename Job” on page 176 for information on creating a VTAM major node rename job.

4. Create the runtime environment transport job.

   The sample transport jobs are created using the RTE Utility option, Generate sample transport JCL. For more information on this option, see “Transporting the Runtime Environment” on page 187.

5. Copy the runtime environment to a remote system using the runtime environment transport job (or any other copy utility).

6. After the copy completes, edit the system variable parameter member RKANPAR(midlev), where the variable midlev is the middle-level qualifier. If necessary, set values for components running on other systems. For example, set values for a hub c running on a different LPAR.

7. Perform the following other remote system set-up tasks as required:

   - Copy the new started tasks to the system procedure library. These started tasks have been enabled for system variables.

   - If you are not using an existing system variable for runtime environment name on the LPAR, set the &SYSNAME system variable to the name of the runtime environment. This is set in SYS1.PARMLIB(IEASYMxx).
Sample Usage Scenario

- If you are using VTAM system variable support, you must run the VTAM major node rename job. This job creates new major nodes that are named according to the system variable specifications. Once the new nodes are created, copy them to SYS1.VTAMLST.
- VSAM file allocation and seeding will most likely be required on every system.

8. Start the component products.
Enabling System Variable Support

Procedure for new and existing runtime environments

You enable system variable support on the second Add Runtime Environment panel. (See “Second Add Runtime Environment Panel” on page 87 for a description of the parameters on that panel.

Some older versions of component products may not support the system variable feature. These component products can exist in the same runtime environment and you will not need to reconfigure these component products in the runtime environment.

Turning on system variable support

Perform the following steps to enable system variable support:

1. From the Main Menu, select Configure products > Select product to configure.

   On the Product Selection Menu, enter S next to the package or component product to select a package or component product to be configured.

   **Note:** Only those packages that are eligible to be configured are listed on this panel. You can only select one product at a time for configuration.

   The Runtime Environments panel displays.

2. Perform one of the following procedures:

   - If you are adding a new runtime environment, perform the steps described in “Adding a Runtime Environment” on page 85. In the process, make sure that you enter the values as instructed in the next steps.

   - If you are updating a runtime environment, do the following:

     A. Enter U (Update) in the Action field next to the runtime environment you are configuring.

        The first Add Runtime Environment panel will display.

     B. Press Enter.

        The second Add Runtime Environment panel will display.

3. Specify the following values to enable system variable support:

   **Table 54. Second Add Runtime Environment Panel System Variable Support**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use OS/390 system variables?</td>
<td>Enter Y to enable support. The default is N.</td>
</tr>
</tbody>
</table>
Important: If you change the status of system variable support in an existing runtime environment (on to off or vice versa), you must reconfigure all component products in that runtime environment. This includes respecifying VTAM values and recreating runtime members.

4. (Optional) Specify the following values on the second Add Runtime Environment panel to enable other functions:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTE name specification</td>
<td>Specify the system variable name by which the runtime environment is identified in the SYS1.PARMLIB LPAR system definition member. The default is &amp;SYSNAME. This value becomes the value of the SYS parameter in all started tasks (for example, SYS='&amp;SYSNAME'). <strong>Note:</strong> Resolved system variable values cannot exceed the length of the variable name (maximum length of 8 characters).</td>
</tr>
<tr>
<td>RTE base alias specification</td>
<td>If this runtime environment is sharing with a base runtime environment, specify an optional system variable specification for the base runtime environment. This value will be inserted into the base runtime environment library references in all started tasks. The resolved name must be a valid library name qualifier. This field is commonly used to switch between base runtime environments at different maintenance levels. You can use the runtime environment base alias as an easy way to switch runtime environment bases and an alternate way to refer to an existing base. <strong>Note:</strong> A label of n/a will be next to this field if the current runtime environment is not sharing with a base runtime environment.</td>
</tr>
<tr>
<td>Applid prefix specification</td>
<td>Specify the VTAM applid prefix that contains system variables. Be sure to place a period after the last symbolic in the specification. The resolved prefix can be a maximum of four characters. The default is K&amp;SYSCLONE.</td>
</tr>
<tr>
<td>Use VTAM model applids?</td>
<td>If you want to use model applids (wildcards), enter Y. Using model applids will generate VTAM nodes that contain applids with wildcard suffixes wherever possible. These wildcards allow usage of any applids that match the pattern within the VTAM node. The default is N.</td>
</tr>
</tbody>
</table>
Creating the System Variable Parameter Member

Procedure

After configuring a runtime environment, you can create the system variable parameter member.

Each runtime environment contains one user system variable parameter member named RKANPAR(midlev), where the variable midlev is the middle-level qualifier. All system-specific values are contained in this member.

Perform the following steps to create the system variable parameter member, midlev, in the RKANPAR library:

1. From the Main Menu, select **Configure products > Select product to configure**. On the Product Selection Menu, enter S next to the package or component product to select a package or component product to be configured.

   **Note:** Only those packages that are eligible to be configured are listed on this panel. You can only select one product at a time for configuration.

   The Runtime Environments panel displays.

2. Enter Z (Utilities) in the Action field next to the runtime environment you are configuring. The RTE Utility panel displays.

3. Select **Create System Variable parameter member**.

   A panel displays a JCL that creates the system variable parameter member, CB#Vxxxx, job.

4. Review the JCL and submit the job.

5. Verify that the job completes successfully. All return codes should be 0.

6. Edit the RKANPAR(PLEX220H) parameter member. Follow the instructions contained in the JCL to ensure proper resolution of cross-system variables.
Creating the VTAM Major Node Rename Job

Procedure

After configuring a runtime environment, you can create the VTAM major node rename job. This job creates VTAM major nodes on remote systems with names that are resolved from the system variable specification.

Perform the following steps to create a VTAM major node rename job in the INSTJOBS library:

1. From the Main Menu, select **Configure products** > **Select product to configure**.
   On the Product Selection Menu, enter $ next to the package or component product to select a package or component product to be configured.

   **Note:** *Only those packages that are eligible to be configured are listed on this panel. You can only select one product at a time for configuration.*

   The Runtime Environments panel displays.

2. Enter Z (Utilities) in the Action field next to the runtime environment you are configuring.
   The RTE Utility panel displays.

3. Select **Create System Variable VTAM major node rename job**.
   A panel displays a JCL that renames the VTAM major node, CB#7xxxx job.

4. Review the JCL. Do not submit the job yet.

   **Note:** *You will submit this job on each remote system where the software is to run. The job is in the RKANSAM library.*
Creating a VTAM Major Node (One Node for All Component Products)

Using a single VTAM major node

A single VTAM major node can contain all the VTAM applids for all of the component products you have configured in the runtime environment. This single major node is then used in place of the individual component product major nodes.

If you choose to use a single VTAM major node, you must create it after all component products have been configured in the runtime environment. Once the node is created and copied to the system VTAM system library (SYS1.VTAMLST), you vary it active and then start all of the component products (started tasks).

Procedure

Perform the following steps to create a single VTAM major node in the RKANSAM library:

1. From the Main Menu, select **Configure products > Select product to configure**.
   
   On the Product Selection Menu, enter S next to the package or component product to select a package or component product to be configured.

   **Note:** *Only those packages that are eligible to be configured are listed on this panel. You can only select one product at a time for configuration.*

   The Runtime Environments panel displays.

2. Enter Z (Utilities) in the Action field next to the runtime environment you are configuring.
   
   The RTE Utility panel displays.

3. Select **Create VTAM major node (one node for all products)**.
   
   The Create VTAM Major Node panel displays.

4. Enter the name you want to use for the single node.
   
   A panel displays a JCL that creates the single node.

5. Review the JCL and submit the job.

6. Verify that the job completed successfully. All return codes should be 0.
Creating a VTAM Major Node (One Node for All Component Products)
Introduction

This chapter provides detailed instructions on using the Configuration tool batch mode processing to do the following:

- Create a new runtime environment by running a single batch job
- Replicate an existing runtime environment
- Transport a replicated runtime environment to other z/OS images

Chapter contents

Using Batch Mode Processing ............................................................ 180
Planning the Runtime Environment Replication Process ...................... 181
Generating Runtime Environment Parameters and Copying to a New Library Member ...... 184
Adding and Configuring Component Product Parameter Decks (optional) ................. 185
Transporting the Runtime Environment ............................................. 187
Using Batch Mode Processing

Configuration tool batch mode processing

Batch mode processing using the Configuration tool is an alternate way of building and configuring an runtime environment. Instead of using the interactive Configuration tool to build and configure a runtime environment, you can submit a single batch job that performs all of the same processing.

Configuration tool batch mode components

The components of batch mode processing in the Configuration tool include the following:

Table 56. Batch Mode Components

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration tool batch job (CICATB)</td>
<td>CICATB is a new job that is generated into the INSTJOBS library. You submit this job to build and configure a runtime environment.</td>
</tr>
<tr>
<td>Configuration tool batch parameter member</td>
<td>This is a single member in INSTJOBS, that contains all of the configuration values for all component products to be configured in the runtime environment.</td>
</tr>
<tr>
<td>RTE Utility to create Configuration tool batch parameter member</td>
<td>This utility creates the batch parameter member for an existing runtime environment, which can then be used for subsequent CICATB job executions. The name of the parameter member is the runtime environment name.</td>
</tr>
</tbody>
</table>

Considerations when creating the Configuration tool batch job

You create the Configuration tool batch job only once on an image. It can then be used for all subsequent batch mode processing on that image.

**Notes:**

- You must recreate the Configuration tool batch job if the ISPF environment has changed.
- When the SUBMIT parameter is set to YES, the generated runtime environment configuration jobs are submitted for execution automatically if the job names do not currently exist in the INSTJOBS library. If the generated jobs already exist, then the jobs are regenerated but not automatically submitted for execution.
- The Job Control Language (JCL) suffix must be unique for each runtime environment, because when the Configuration tool batch job runs, it is used in every member name that is generated in INSTJOBS. If the suffix is not unique, the jobs that are generated will conflict with other runtime environment jobs that may already exist in INSTJOBS.
Planning the Runtime Environment Replication Process

Batch mode processing in the Configuration tool is tailored for installations that want to replicate an existing runtime environment onto any number of z/OS images.

Once you have interactively created and configured an runtime environment for the component products, the Configuration tool batch mode processing option allows you to collect those parameters and replicate the runtime environment onto any local or remote z/OS image.

The Configuration tool batch mode process involves the following three general steps:

- Creating batch mode parameters
- Adding and configuring parameter input decks
- Creating the new runtime environment on the appropriate z/OS image, using the appropriate runtime environment transportation method

**Note:** Once a runtime environment has been created in batch mode, it can later be accessed and further configured using the interactive Configuration tool.

Creating batch mode parameters

In this step, you invoke the Create batch mode parameters processing option to export an existing runtime environment’s parameters into a library member.

You then copy the member and change the image-specific parameters, as necessary, to configure the runtime environment for its new environment.

Adding and configuring parameter input decks (optional)

In this step, you edit the newly created member and invoke Configuration tool edit macros to add additional component product parameter decks to it.

Once you have finished adding component products to the member, configure the component product by changing the default parameter values as required.
Creating a new runtime environment on the appropriate z/OS image using the appropriate transport method

In this step, you create the new runtime environment on the appropriate z/OS image using one of the following transport methods:

<table>
<thead>
<tr>
<th>Table 57. Runtime Environment Transport Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Transport Method</strong></td>
</tr>
</tbody>
</table>
| Define runtime environment on local z/OS image using shared DASD | Interactive and batch mode Configuration tool on the local z/OS image, to create a runtime environment accessible to the target image. | The following are advantages of this method:  
- The interactive Configuration tool, located on the local image, contains the configuration information for all images.  
- Only one copy of the runtime libraries is created.  
- Only one batch job is submitted.  
The disadvantage of using this method is that this is only applicable for z/OS images with shared DASD. |
| Transport runtime environment from local image to remote image | Interactive and batch mode Configuration tool on the local image to create a runtime environment. Once the runtime environment has been defined, you use sample transport jobs to ship the runtime libraries and parameters to the remote image. | The following are advantages of this method:  
- The interactive Configuration tool, located on the local image, contains the configuration information for all images.  
- Only one batch job is submitted.  
- This is applicable for remote z/OS images that do not share DASD.  
The disadvantage of using this method is that two copies of the runtime libraries are created. |
| Transport runtime environment batch jobs from local z/OS image to remote image | Interactive and batch mode Configuration tool on the local image to create a set of batch jobs that can build a runtime environment. Once created, you use sample transport jobs to ship the batch jobs to the remote image. The jobs are manually submitted on the remote image to create the runtime libraries and parameters. | The following are advantages of this method:  
- The interactive Configuration tool, located on the local image, contains the configuration information for all images.  
- This is applicable for remote z/OS images that do not share DASD.  
- Only one copy of the runtime libraries is created.  
The disadvantage of using this method is that you must manually submit a series of batch jobs, or use the Auto Submit CLIST to submit the jobs that will create the runtime environment. |
Planning the Runtime Environment Replication Process

Table 57. Runtime Environment Transport Methods (continued)

<table>
<thead>
<tr>
<th>Transport Method</th>
<th>Methods Used</th>
<th>Advantages &amp; Disadvantages</th>
</tr>
</thead>
</table>
| Transport runtime environment batch mode parameters from local z/OS image to remote image equipped with the Configuration tool | Interactive Configuration tool on the local image to export an existing runtime environment.  
Once the runtime environment parameters have been collected, you use the sample transport jobs to ship the batch mode parameters to the remote image.  
The batch mode Configuration tool is run on the remote image to create the runtime libraries and parameters. | The following are advantages of this method:  
- This is applicable for remote z/OS images that do not share DASD.  
- Only one copy of the runtime libraries is created.  
- Only one batch job is submitted.  
The disadvantage of using this method is that the interactive Configuration tool located on the local image does not contain the configuration information for all images. |
Generating Runtime Environment Parameters and Copying to a New Library Member

Procedure

You can generate parameter decks for all component products within an existing runtime environment, and then copy the information into a new library member to be used during batch mode processing.

Perform the following steps to generate the runtime environment parameters and copy the information into a new library member:

1. From the Main Menu, select **Configure products > Select product to configure**.
   On the Product Selection Menu, enter S next to the package or component product to select a package or component product to be configured.

   *Note: Only those packages that are eligible to be configured are listed on this panel. You can only select one product at a time for configuration.*

   The Runtime Environments panel displays.

2. Enter Z (Utilities) in the Action field next to the runtime environment you are configuring.
   The RTE Utility panel displays.

3. Select **Create batch mode parameters**.
   The Create Batch Mode Parameters panel displays.

4. Specify the library that will receive the batch parameter member generated by the Configuration tool.
   The INSTJOBS library is specified by default, and the member name will be the same as that of the current runtime environment.

5. Exit the Configuration tool.

6. Edit the INSTJOBS library and copy the exported library member to a new member name.
   This new member name will also be used as the name of the new runtime environment.

7. Using ISPF Option 2, edit the new library member to reflect the settings specific to the z/OS image where the new runtime environment will exist.
Adding and Configuring Component Product Parameter Decks
(optional)

Procedure

You can add parameter decks for additional component products to the newly created library member.

Perform the following steps to add and configure parameter decks for additional component products:

1. Start the Configuration tool on the local image.
2. From the Main Menu, select **Configure Products > Services and utilities > Create batch mode job**.

   **Note:** *KCISETUP needs to be created once on an image and can be used for all subsequent parameter deck processing on that image. KCISETUP must be recreated if the ISPF environment has changed, or you subsequently split the INSTLIB.*

3. Exit the Configuration tool.
4. Set the environmental variables required to run the Configuration tool edit macros. Do the following:
   
   1. Log onto a TSO session, invoke ISPF, and then go to TSO command mode (Option 6 in ISPF).
   2. Enter the following command:
   
   ```
   EX 'shilev.INSTLIB(KCISETUP)'
   ```
   
   where *shilev* is the SMP/E dataset high-level qualifier.

5. Using ISPF Option 2, edit the library member that was created using “Generating Runtime Environment Parameters and Copying to a New Library Member” on page 184, entering the following on the command line:

   - For System Modification Program/Extended (SMP/E) installs, enter the following:
   
   ```
   KCICPGEN Kppvvv
   ```
   
   where the following are the variables:

   - **pp** = the two-letter component code for the component product you are configuring
   - **vvv** = the version of that component product

   **Note:** *Only supply the component code and version that identifies the package you purchased. Underlying components that make up the package will be included.*

   Repeat the above command until a parameter deck is generated for all the component products you want to add.

   - For a Quick Install, enter the following:
   
   ```
   KCICPGEN QI
   ```
The KCICPGEN command invokes the Configuration tool edit macro, which uses the Kppuvv or QI parameter within the command string to locate the correct component product information (PI) member within INSTDATA. Once it finds the PI member, the edit macro scans the file and builds a list of all components and parameters required for the specified component product. A component product that already exists within the library member is not replaced.

6. Change the component product parameter values within the library member as required. The values displayed represent default settings.

7. Once you have finished editing the library member, save the changes. The library member is now ready for batch processing.
Transporting the Runtime Environment

Using the appropriate runtime environment transport method

Use any of the following transport methods to create the new runtime environment on the appropriate z/OS image:

- Define a runtime environment on a local z/OS image using shared DASD
- Transport a runtime environment from a local z/OS image to a remote image
- Transport runtime environment batch jobs from a local z/OS image to a remote image
- Transport runtime environment batch mode parameters from a local z/OS image to a remote image

Before using the transport methods within this section, make sure that sufficient space and library security authorizations exist.

For a list of the advantages and disadvantages for each transport method see “Creating a new runtime environment on the appropriate z/OS image using the appropriate transport method” on page 182.

Defining a runtime environment on a local z/OS image using shared direct access storage device

Perform the following steps to define a runtime environment on a local z/OS image using shared DASD:

1. Start the Configuration tool on the local image.
2. From the Main Menu, select Configure Products > Services and utilities > Create batch mode job.
3. Exit the Configuration tool.
4. Perform a scan on the runtime environment parameters. Do the following:
   1. Edit CICATB, updating the BATCHLIB and BATCHMEM parameters as necessary, and setting the SUBMIT parameter to SCAN.
   2. Submit the CICATB job to scan the runtime environment parameters.
   3. Verify that the job completes successfully. Do the following:
      A. Review the parameter report.
      B. Correct any errors in the parameter member.
      C. Repeat the scan until a clean report is generated.
5. Create a new runtime environment that is accessible to the target image. Do the following:
   1. Edit CICATB again, setting the SUBMIT parameter to YES.
      This submits the runtime environment configuration jobs that allocate and populate runtime libraries.
   2. Submit the CICATB job to create the runtime environment.
3. Verify that the job completes successfully.

6. Perform any manual configuration steps on the target image, such as the following:
   - Copying procedures to PROCLIB
   - Copying VTAM definitions to VTAMLST
   - APF-authorizing libraries

**Transporting runtime environment from a local z/OS image to a remote image**

Perform the following steps to transport a runtime environment from a local z/OS image to a remote image:

1. Start the Configuration tool on the local image.
2. From the Main Menu, select **Configure Products > Services and utilities > Create batch mode job**.
3. Exit the Configuration tool.
4. Perform a scan on the runtime environment parameters. Do the following:
   1. Edit CICATB, updating the BATCHLIB and BATCHMEM parameters as necessary, and setting the SUBMIT parameter to `SCAN`.
   2. Submit the CICATB job to scan the runtime environment parameters.
   3. Verify that the job completes successfully. Do the following:
      - A. Review the parameter report.
      - B. Correct any errors in the parameter member.
      - C. Repeat the scan until a clean report is generated.
5. Create a new runtime environment that is accessible to the target image runtime environment. Do the following:
   1. Edit CICATB again, setting the SUBMIT parameter to `YES`.
      - This submits the runtime environment configuration jobs that allocate and populate runtime libraries.
   2. Submit the CICATB job to create the runtime environment.
   3. Verify that the job completes successfully.
6. Start the Configuration tool again.
7. From the Main Menu, select **Configure products > Select product to configure**.
   On the Product Selection Menu, enter `s` next to the package or component product to select a package or component product to be configured.

**Note:** Only those packages that are eligible to be configured are listed on this panel. You can only select one product at a time for configuration.

The Runtime Environments panel displays.

8. Enter `Z` (Utilities) in the Action field next to the runtime environment you are configuring.
   The RTE Utility panel displays.
9. Select **Generate sample transport JCL**

   This action will cause several sample transport jobs to be generated within the RKANSAM library. Member $XPRTNDX provides a description of all generated members.

   For example, to use DFDSS to transport the runtime libraries to the target image, use one of the following sample jobs:
   - XDFDMP01 on the master image to dump the runtime libraries
   - XDFRST01 on the target image to restore the runtime libraries

   You have now successfully transported a new runtime environment to the target image.

10. Perform any manual configuration steps on the target image, such as the following:
   - Copying procedures to PROCLIB
   - Copying VTAM definitions to VTAMLST
   - APF-authorizing libraries

**Transporting runtime environment batch jobs from a local z/OS image to a remote image equipped with the Configuration tool**

   Perform the following steps to transport runtime environment batch jobs from a local z/OS image to a remote image that is equipped with the Configuration tool:

   1. Start the Configuration tool on the local image.
   2. From the Main Menu, select **Configure Products > Services and utilities > Create batch mode job**.
   3. Exit the Configuration tool.
   4. Perform a scan on the runtime environment parameters. Do the following:
      1. Edit CICATB, updating the BATCHLIB and BATCHMEM parameters as necessary, and setting the SUBMIT parameter to SCAN.
      2. Submit the CICATB job to scan the runtime environment parameters.
      3. Verify that the job completes successfully. Do the following:
         A. Review the parameter report.
         B. Correct any errors in the parameter member.
         C. Repeat the scan until a clean report is generated.
   5. Create the runtime environment generation jobs:
      1. Edit CICATB again, setting the SUBMIT parameter to NO.
         This creates the runtime environment configuration jobs that allocate and populate runtime libraries.
      2. Submit the CICATB job to create the runtime environment generation jobs.
      3. Verify that the job completes successfully.

   You have now successfully created a set of runtime environment configuration batch jobs that must be transported to the target image.
6. Start the Configuration tool again.

7. From the Main Menu, select **Configure products > Select product to configure.**
   On the Product Selection Menu, enter $ next to the package or component product to select a package or component product to be configured.

   **Note:** *Only those packages that are eligible to be configured are listed on this panel. You can only select one product at a time for configuration.*

   The Runtime Environments panel displays.

8. Enter Z (Utilities) in the Action field next to the runtime environment you are configuring.
   The RTE Utility panel displays.

9. Select **Generate sample transport JCL.**
   This will cause several sample transport jobs to be generated within the RKANSAM library. Member $XPRTNDX provides a description of all generated members.

   For example, to use DFDSS to transport the targets, INSTLIB, INSTDATA, and INSTJOBS to the remote image, use of the following sample jobs:
   - XDFDMP03 on the master image to dump the batch jobs
   - XDFRST03 on the target image to restore the batch jobs

10. Submit the batch jobs on the target image in the order listed in the Jobs Sorted By Generation Sequence section of the Configuration tool batch mode job report.

    You can submit each job manually or use the Auto Submit CLIST to automatically submit the Configuration tool jobs on the target image. To use the Auto Submit CLIST, do the following:

    1. Verify that the SMP/E target libraries are available on this image upon which the CLIST will be executed.
    2. Edit the member named SUB#jclsuffix in INSTJOBS, where the variable jclsuffix identifies the JCL suffix for the new runtime environment.
    3. Execute the CLIST to submit the Configuration tool jobs that will create the runtime environment.

    You have successfully created a new runtime environment on the target image.

11. Perform any manual configuration steps on the target image, such as the following:
   - Copying procedures to PROCLIB
   - Copying VTAM definitions to VTAMLST
   - APF-authorizing libraries

**Transporting runtime environment batch mode parameters from a local z/OS image to a remote image**

Perform the following steps to transport runtime environment batch mode parameters from a local z/OS image to a remote image:

1. Start the Configuration tool on the local image.
2. From the Main Menu, select **Configure products > Select product to configure.**

   On the Product Selection Menu, enter $ next to the package or component product to select a package or component product to be configured.

   **Note:** *Only those packages that are eligible to be configured are listed on this panel. You can only select one product at a time for configuration.*

   The Runtime Environments panel displays.

3. Enter Z (Utilities) in the Action field next to the runtime environment you are configuring.

   The RTE Utility panel displays.

4. Select **Generate sample transport JCL.**

   This will cause several sample transport jobs to be generated within the RKANSAM library. Member $XPRTNDX provides a description of all generated members.

   For example, to use DFDSS to transport the targets, INSTLIB, INSTDATA, and INSTJOBS to the remote image, use one of the following sample jobs:
   - XDFDMP03 on the master image to dump the batch jobs
   - XDFRST03 on the target image to restore the batch jobs

   You have successfully transported the batch mode parameters to the target image.

5. From the Main Menu, select **Configure Products > Services and utilities > Create batch mode job.**

6. Exit the Configuration tool.

7. Perform a scan on the runtime environment parameters. Do the following:
   1. Edit CICATB, updating the BATCHLIB and BATCHMEM parameters as necessary, and setting the SUBMIT parameter to SCAN.
   2. Submit the CICATB job to scan the runtime environment parameters.
   3. Verify that the job completes successfully. Do the following:
      - Review the parameter report.
      - Correct any errors in the parameter member.
      - Repeat the scan until a clean report is generated.

8. Create the runtime environment on the target image. Do the following:
   1. Edit CICATB again, setting the SUBMIT parameter to YES.
      - This submits the runtime environment configuration jobs that allocate and populate runtime libraries.
   2. Submit the CICATB job to create the runtime environment.
   3. Verify that the job completes successfully.

9. Perform any manual configuration steps on the target image, such as the following:
   - Copying procedures to PROCLIB
   - Copying VTAM definitions to VTAMLST
Transporting the Runtime Environment

- APF-authorizing libraries
Introduction

This appendix provides a series of scenarios that IBM recommends for configuring component products.

Appendix contents

Using the Configuration Scenarios ................................................................. 194
Scenario 1: Rolling Out Component Products Across a 40 LPAR Enterprise
(Standard Method) ......................................................................................... 195
Scenario 2: Rolling Out Component Products Across a 40 LPAR Enterprise (Common
RKANPAR and Common RKANCMD Libraries) ............................................... 198
Scenario 3: Replicating a Runtime Environment to Another LPAR Using Configuration
Tool Batch Mode .......................................................................................... 201
Scenario 4: Replicating a Runtime Environment to Another LPAR Using Configuration
Tool Batch Auto-submit ................................................................................ 203
Scenario 5: Replicating a Runtime Environment on Same LPAR Using Batch Processing .... 206
Scenario 6: Replicating a Runtime Environment Using Interactive Copy ......................... 208
Scenario 7: Copying Configuration Values from One Configuration Tool to Another
Configuration Tool ....................................................................................... 209
Using the Configuration Scenarios

Location of supporting topics with detailed procedures

The following table lists the supporting topics with detailed procedures in this manual and indicates which topic applies to each scenario:

<table>
<thead>
<tr>
<th>Supporting Topics with Detailed Procedures</th>
<th>Scenario 1</th>
<th>Scenario 2</th>
<th>Scenario 3</th>
<th>Scenario 4</th>
<th>Scenario 5</th>
<th>Scenario 6</th>
<th>Scenario 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installing component products</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Note:</strong> For instructions on installing component products, refer to the “Program Directory” for the component product.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Configure component products in runtime environments</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Creating the System Variable Parameter Member</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Creating VTAM Major Node (one node for all component products)</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KCICPGEN: generating the Batch Parameter Input Deck</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create, replicate, &amp; transport runtime environments (batch mode processing)</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transport runtime environment batch mode parameters from a local z/OS image to a remote image</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transport runtime environment batch jobs from a local z/OS image to a remote image equipped with the Configuration tool</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Scenario 1: Rolling Out Component Products Across a 40 LPAR Enterprise (Standard Method)

This scenario describes a standard Configuration tool method for rolling out component products across the z/OS enterprise. You install and configure all component products on the main LPAR, and then copy items from the main LPAR to other systems and LPARs throughout the enterprise.

**Note:** This particular scenario shows 40 LPARs. You can use this scenario for any number of LPARs at the site.

This scenario assumes that you are using the following:

- IBM symbolics on all the LPARs
- Consistent naming conventions across all of the LPARs (dataset names, started tasks, and VTAM applids)

Pros and Cons of using this scenario

Table 59. Pros and Cons of Scenario 1

<table>
<thead>
<tr>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Only 3 unique runtime environments need to be configured</td>
<td>Cannot update runtime environments on other LPARs with the Configuration tool</td>
</tr>
<tr>
<td>Runtime environments are copied to other LPARs</td>
<td></td>
</tr>
<tr>
<td>1 Configuration tool</td>
<td></td>
</tr>
<tr>
<td>Component products use symbolic values defined on LPAR they run on</td>
<td></td>
</tr>
<tr>
<td>Fast rollout across enterprise</td>
<td></td>
</tr>
<tr>
<td>Consistency</td>
<td></td>
</tr>
<tr>
<td>Potential DASD savings</td>
<td></td>
</tr>
</tbody>
</table>

Procedure

To use this scenario, perform the following steps:

1. Install all the component products on Central Processor Complex (CPC) A, LPAR A1 (Main). Refer to the Program Directory for detailed installation instructions. This sets up one Configuration tool and one System Modification Program/Extended (SMP/E) environment that will be used to support all LPARs.

2. Create the RTE BASE on LPAR A1 (Main).

3. Create the SHARING runtime environment template on LPAR A1 (Main).
   This runtime environment shares the RTE BASE, has system variables enabled, and is a superset configuration of what will run on other LPARs.

**Note:** If you are using a hub and remote Candle Management Server (CMS), create a hub SHARING runtime environment template and a remote SHARING runtime environment template.
4. After configuring the SHARING runtime environment template, do the following:
   1. Create the system variable member.
   2. (Optional) Create the single VTAM major node. This will simplify VTAM set up.

5. Copy the RTE BASE datasets to LPAR B1 (Target), using any copy or rename method or the supplied Configuration tool transport runtime environment batch jobs. Keep the same dataset names, if possible.

6. Copy and rename the SHARING runtime environment template datasets to each execution LPAR, using any copy/rename method or the supplied Configuration tool transport runtime environment batch jobs. As the files are copied, rename them to match the execution LPAR name.

7. In each RKANPAR library, manually create a new system variable member that matches the name of the execution LPAR by copying the contents of the system variable member you created in step 4A. If necessary, in this new member change LPAR-specific values and update cross-system values for component products running on different LPARs.

8. Set up the component products on all other systems and LPARs in the enterprise.

9. Repeat steps 5. to 8. for each system that exists in the enterprise.

10. Complete the configuration. At a minimum you must do the following:
    - Copy the started tasks and VTAM major nodes to the system libraries.
    - APF-authorize the datasets on the appropriate LPARs.

11. Start each of the component products.
Scenario 1: Rolling Out Component Products Across a 40 LPAR Enterprise (Standard Method)

Figure 12. Two Systems with 6 LPARs in Scenario 1

<table>
<thead>
<tr>
<th>CPC A</th>
<th>CPC B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shared DASD within CPC A</td>
<td>Shared DASD within CPC B</td>
</tr>
<tr>
<td>1 Configuration tool and 1 SMP/E environment</td>
<td>1 copy of RTE BASE</td>
</tr>
<tr>
<td>1 RTE BASE</td>
<td>3 copies SHARING RTE A1 template</td>
</tr>
<tr>
<td>1 SHARING RTE A1 template</td>
<td>2 copies SHARING RTE A1 template</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LPAR A1 (Main)</th>
<th>LPAR A2</th>
<th>LPAR A3</th>
<th>LPAR A4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration tool, SMP/E, Target libraries</td>
<td>RTE BASE</td>
<td>&lt;copy of SHARING RTE A1 renamed to A3&gt;</td>
<td>&lt;copy of SHARING RTE A1 renamed to A4&gt;</td>
</tr>
<tr>
<td>SHARING RTE A1 template</td>
<td>Rikanpar member A1</td>
<td>Rikanpar member A3</td>
<td>Rikanpar member A4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LPAR B1 (Target)</th>
<th>LPAR B2</th>
<th>LPAR B3</th>
<th>LPAR B4</th>
<th>LPAR B5</th>
<th>LPAR B6</th>
<th>LPAR B7</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;RTE BASE&gt;</td>
<td></td>
<td></td>
<td>&lt;copy of SHARING RTE A1 renamed to B1&gt;</td>
<td>&lt;copy of SHARING RTE A1 renamed to B4&gt;</td>
<td>&lt;copy of SHARING RTE A1 renamed to B7&gt;</td>
<td></td>
</tr>
<tr>
<td>RKANPAR member B1</td>
<td>RKANPAR member B2</td>
<td>RKANPAR member B3</td>
<td>RKANPAR member B4</td>
<td>RKANPAR member B5</td>
<td>RKANPAR member B6</td>
<td></td>
</tr>
</tbody>
</table>

Legend: <item> = Copied from main LPAR A1
Scenario 2: Rolling Out Component Products Across a 40 LPAR Enterprise (Common RKANPAR and Common RKANCMD Libraries)

This scenario describes another method for rolling out component products across the z/OS enterprise. In the scenario you create a new common RKANPAR library and a new common RKANCMD library, that will contain all of the unique configuration values required by all of the LPARs. You install and configure all component products on the main LPAR, and then copy items from the main LPAR to other LPARs throughout the enterprise.

Note: This particular scenario shows 40 LPARs. You can use this scenario for any number of LPARs at the site.

This scenario assumes that you are using the following:

- IBM symbolics on all the LPARs
- Consistent naming conventions across all of the LPARs (dataset names, started tasks, and VTAM applids)

Pros and Cons of using this scenario

Table 60. Pros and Cons of Scenario 2

<table>
<thead>
<tr>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 common RKANPAR and RKANCMD library for all LPARs</td>
<td>Cannot update runtime environments on other LPARs with the Configuration tool</td>
</tr>
<tr>
<td>Only 3 unique runtime environments need to be configured</td>
<td>Manual updates are required</td>
</tr>
<tr>
<td>Runtime environments are copied to other LPARs</td>
<td></td>
</tr>
<tr>
<td>1 Configuration tool</td>
<td></td>
</tr>
<tr>
<td>Component products use symbolic values defined on LPAR they run on</td>
<td></td>
</tr>
<tr>
<td>Fast rollout across enterprise</td>
<td></td>
</tr>
<tr>
<td>Consistency</td>
<td></td>
</tr>
<tr>
<td>Potential DASD savings</td>
<td></td>
</tr>
</tbody>
</table>

Procedure

To use this scenario, perform the following steps:

1. Install all the component products on Central Processor Complex (CPC) A, LPAR A1 (Main). Refer to the Program Directory for detailed installation instructions.
   This sets up one Configuration tool and one SMP/E environment that will be used to support all LPARs.

2. Create the RTE BASE on LPAR A1 (Main).

3. Create the SHARING runtime environment template on LPAR A1 (Main).
This runtime environment shares the RTE BASE, has system variables enabled, and is a superset configuration of what will run on other LPARs.

**Note:** If you are using a hub and remote CMS, create a hub SHARING runtime environment template and a remote SHARING runtime environment template.

4. After configuring the SHARING runtime environment template, do the following:
   1. Create the system variable parameter member. Follow the instructions in “Creating the System Variable Parameter Member” on page 175.
   2. Create common RKANPAR and RKANCMD datasets and then copy the runtime environment template datasets into these common datasets.
   3. In the common RKANPAR, create additional system variable members as necessary. The names of these members match the other LPAR names. For each member, change LPAR-specific values as necessary. Review cross-system values for component products on different LPARs.
   4. Edit RKANSAM started tasks and change all occurrences of the RKANPAR/RKANCMD dataset names to the common dataset names. For example:
      ```
      C '&RHILEV..&SYS..RKANPAR' 'common.RKANPAR' all
      C '&RHILEV..&SYS..RKANCMD' 'common.RKANCMD' all
      ```
   5. (Optional) Create the single VTAM major node. Follow the instructions in “Creating a VTAM Major Node (One Node for All Component Products)” on page 177.
      This will simplify VTAM set up.

5. Copy the following to LPAR B1 (Target), using any copy/rename method or the supplied Configuration tool transport runtime environment batch jobs:
   - RTE BASE datasets
   - Common RKANPAR and RKANCMD datasets
   Keep the same dataset names, if possible.

6. Copy and rename the SHARING runtime environment template datasets to each execution LPAR, using any copy/rename method or the supplied Configuration tool transport runtime environment batch jobs. As the files are copied, rename them to match the execution LPAR name.

7. Set up the component products on all other systems and LPARs in the enterprise. Repeat steps 5-6 for each system that exists in the enterprise.

   The following diagram shows 2 systems that contain 6 LPARs.

8. Complete the configuration. At a minimum you must do the following:
   - Copy the started tasks and VTAM major nodes to the system libraries.
   - APF-authorize the datasets on the appropriate LPARs.

9. Start each of the component products.
### Scenario 2: Rolling Out Component Products Across a 40 LPAR Enterprise (Common RKANPAR and Common RKANCMD Libraries)

#### Figure 13. Two Systems with 6 LPARs in Scenario 2

<table>
<thead>
<tr>
<th>CPC A</th>
<th>CPC B</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Shared DASD within CPC A</strong></td>
<td><strong>Shared DASD within CPC B</strong></td>
</tr>
<tr>
<td>1 Configuration tool and 1 SMP/E environment</td>
<td>1 copy of RTE BASE</td>
</tr>
<tr>
<td>1 RTE BASE</td>
<td>3 copies of SHARING RTE A1 template</td>
</tr>
<tr>
<td>1 SHARING RTE A1 template</td>
<td>1 Common RKANPAR and RKANCMD</td>
</tr>
<tr>
<td>2 copies of SHARING RTE A1 template</td>
<td></td>
</tr>
<tr>
<td>1 Common RKANPAR and RKANCMD</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LPAR A1 (Main)</th>
<th>LPAR A3</th>
<th>LPAR A4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration tool, SMP/E, Target libraries</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RTE BASE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SHARING RTE A1 template</td>
<td>&lt;copy of SHARING RTE A1 renamed to A3&gt;</td>
<td>&lt;copy of SHARING RTE A1 renamed to A4&gt;</td>
</tr>
<tr>
<td>Common RKANPAR RKANCMD with members A1, A3, A4, B1, B4, B7</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LPAR B1 (Target)</th>
<th>LPAR B4</th>
<th>LPAR B7</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;RTE BASE&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;copy of SHARING RTE A1 renamed to B1&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;Common RKANPAR RKANCMD with members A1, A3, A4, B1, B4, B7&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;copy of SHARING RTE A1 renamed to B4&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;copy of SHARING RTE A1 renamed to B7&gt;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Legend:** *<item> = Copied from main LPAR A1*
Scenario 3: Replicating a Runtime Environment to Another LPAR Using Configuration Tool Batch Mode

This scenario describes a method for creating a runtime environment on another LPAR using batch mode processing in the Configuration tool.

Pros and Cons of using this scenario

<table>
<thead>
<tr>
<th>Table 61. Pros and Cons of Scenario 3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pros</strong></td>
</tr>
<tr>
<td>- Less interactive Configuration tool work</td>
</tr>
<tr>
<td>- Runtime environment creation with a single batch job</td>
</tr>
<tr>
<td>- Runtime environment detail report</td>
</tr>
<tr>
<td><strong>Cons</strong></td>
</tr>
<tr>
<td>- None</td>
</tr>
</tbody>
</table>

Procedure

To use this scenario, perform the following steps:

1. Install all the component products on Central Processor Complex (CPC) A, LPAR A1 (Main). Refer to the Program Directory for detailed installation instructions. This sets up one Configuration tool and one SMP/E environment that will be used to support all LPARs.

2. Copy the Configuration tool and the SMP/E target libraries to CPC B, using any copy or rename method or the supplied Configuration tool transport runtime environment batch jobs.

3. Create the RTE BASE on LPAR A1 (Main).

4. Create SHARING runtime environment A1 on LPAR A1 (Main). This runtime environment shares the RTE BASE.

5. Create the Configuration tool batch parm deck for SHARING runtime environment A1.

6. Create a new batch parm deck for each of the other LPARs (A3, A4, B1, B4, and B7, as shown in the following diagram), by copying the contents of the SHARING runtime environment A1 batch parm deck.

7. In each new batch parm deck, review and modify the values accordingly.

8. Submit the Configuration tool batch job on CPC A for each LPAR in CPC A (A3 and A4). This creates the required runtime environments for CPC A.

9. Copy the batch parm decks for each LPAR in CPC B (B1, B4, B7) to the Configuration tool on CPC B.

10. Submit the Configuration tool batch job on CPC B for each LPAR in CPC B (B1, B4, B7). This creates the required runtime environments for CPC B.

11. Copy the RTE BASE datasets to the LPAR B1 (Target). Keep the same dataset names, if possible.
12. Complete the configuration. At a minimum you must the following:

- Copy the started tasks and VTAM major nodes to the system libraries.
- APF-authorize the datasets on the appropriate LPARs.

13. Start each of the component products.

**Figure 14. Two Systems with 6 LPARs in Scenario 3**

<table>
<thead>
<tr>
<th>CPC A</th>
<th>CPC B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shared DASD within CPC A</td>
<td>Shared DASD within CPC B</td>
</tr>
<tr>
<td>1 Configuration tool and 1 SMP/E environment (includes Target libraries)</td>
<td>1 Configuration tool</td>
</tr>
<tr>
<td>1 RTE BASE</td>
<td>1 Copy of RTE BASE</td>
</tr>
<tr>
<td>3 Unique RTEs</td>
<td>3 Unique RTEs</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LPAR A1 (Main)</th>
<th>LPAR A3</th>
<th>LPAR A4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration tool, SMP/E, Target libraries</td>
<td>RTE BASE</td>
<td>SHARING RTE A1</td>
</tr>
<tr>
<td>(RTE A3)</td>
<td>(RTE A4)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LPAR B1 (Target)</th>
<th>LPAR B4</th>
<th>LPAR B7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration tool, &lt;Target libraries&gt;</td>
<td>&lt;RTE BASE&gt;</td>
<td>&lt;RTE B1&gt;</td>
</tr>
<tr>
<td>(RTE B1)</td>
<td>(RTE B4)</td>
<td>(RTE B7)</td>
</tr>
</tbody>
</table>

**Legend:**

- `<item>` = Copied from main LPAR A1
- `{item}` = Created with Configuration tool Batch
Scenario 4: Replicating a Runtime Environment to Another LPAR Using Configuration Tool Batch Auto-submit

This scenario describes a method for creating a runtime environment on another LPAR using the Auto-submit feature of batch mode processing in the Configuration tool.

This is similar to Scenario 4, except that the runtime environments are created by canned Job Control Language (JCL) built on the main LPAR using Configuration tool batch mode processing. The JCL is copied to the execution LPAR and submitted by the Auto Submit CLIST.

Pros and Cons of using this scenario

Table 62. Pros and Cons of Scenario 4

<table>
<thead>
<tr>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Configuration tool contains all runtime environment configuration values</td>
<td>None</td>
</tr>
<tr>
<td>Runtime environment creation with a single batch job</td>
<td></td>
</tr>
<tr>
<td>Runtime environment detail report</td>
<td></td>
</tr>
<tr>
<td>All JCL built on the main LPAR</td>
<td></td>
</tr>
<tr>
<td>Less interactive Configuration tool work</td>
<td></td>
</tr>
</tbody>
</table>

Procedure

To use this scenario, perform the following steps:

1. Install all the component products on Central Processor Complex (CPC) A, LPAR A1 (Main). Refer to the Program Directory for detailed installation instructions. This sets up one Configuration tool and one SMP/E environment that will be used to support all LPARs.

2. Copy the SMP/E target libraries to CPC B, using any copy/rename method or the supplied Configuration tool transport runtime environment batch jobs.

3. Create the RTE BASE on LPAR A1 (Main).

4. Create SHARING runtime environment A1 on LPAR A1 (Main). This runtime environment shares the RTE BASE.

5. Create the Configuration tool batch parm deck for SHARING runtime environment A1.

6. Create a new batch parm deck for each of the other LPARs (A3, A4, B1, B4, and B7, as shown in the following diagram), by copying the contents of the SHARING runtime environment A1 batch parm deck.

7. In each new batch parm deck, review and modify the values accordingly.
8. On CPC A do the following:
   
   1. Submit the Configuration tool batch job for each LPAR in CPC A (A3 and A4), and set the SUBMIT parameter to YES in the batch job options.
      This creates the required runtime environments for CPC A.
   
   2. Submit the Configuration tool batch job for each LPAR in CPC B (B1, B4, B7) and set the SUBMIT parameter to NO in the batch job options.
      This creates the JCL necessary for creating the runtime environment on each execution LPAR in CPC B.

9. Copy the JCL members from INSTJOBS to CPC B.

   Note: These members all end in the same 4-character suffix that you specified in the Configuration tool batch parm deck.

10. On CPC B, execute the Auto Submit CLIST for each LPAR in CPC B.
    
    This submits the jobs required to create the runtime environments for CPC B (B1, B4, B7).

11. Copy the RTE BASE datasets to the LPAR B1 (Target). Keep the same dataset names, if possible.

12. Complete the configuration. At a minimum you must do the following:

   - Copy the started tasks and VTAM major nodes to the system libraries.
   - APF-authorize the datasets on the appropriate LPARs.

13. Start each of the component products.
**Figure 15. Two Systems with 6 LPARs in Scenario 4**

<table>
<thead>
<tr>
<th>CPC A</th>
<th>CPC B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shared DASD within CPC A</td>
<td></td>
</tr>
<tr>
<td>1 Configuration tool and 1 SMP/E environment (includes Target libraries)</td>
<td></td>
</tr>
<tr>
<td>1 RTE BASE</td>
<td></td>
</tr>
<tr>
<td>3 Unique RTEs</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LPAR A1 (Main)</th>
<th>LPAR A3</th>
<th>LPAR A4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration tool, SMP/E, Target libraries</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RTE BASE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SHARING RTE A1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>{JCL to create RTEs A3, A4, B1, B4, B7}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;JCL to create RTE A3&gt;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LPAR B1 (Target)</th>
<th>LPAR B4</th>
<th>LPAR B7</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;Target libraries&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;RTE BASE&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>{RTE B1}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;JCL to create RTE B1&gt;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Legend: `<item>` = Copied from main LPAR A1  
`{item}` = Created with Configuration tool Batch
Scenario 5: Replicating a Runtime Environment on Same LPAR Using Batch Processing

This scenario describes a method for replicating a runtime environment on the same LPAR using batch mode processing in the Configuration tool. You install and configure all component products on the main LPAR, and then copy the runtime environments to the appropriate LPARs.

Pros and Cons of using this scenario

Table 63. Pros and Cons of Scenario 5

<table>
<thead>
<tr>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Configuration tool</td>
<td>Duplicated runtime environments temporarily use more DASD</td>
</tr>
<tr>
<td>1 set of Target libraries</td>
<td></td>
</tr>
<tr>
<td>Configuration tool batch</td>
<td></td>
</tr>
<tr>
<td>More copies</td>
<td></td>
</tr>
</tbody>
</table>

Procedure

To use this scenario, perform the following steps:

1. Install all the component products on Central Processor Complex (CPC) A, LPAR A1 (Main). Refer to the Program Directory for detailed installation instructions. This sets up one Configuration tool and one SMP/E environment that will be used to support all LPARs.

2. Create the RTE BASE on LPAR A1 (Main).

3. Create SHARING runtime environment A1 on LPAR A1 (Main). This runtime environment shares the RTE BASE.


5. Create a new batch parm deck for each of the other LPARs (A3, A4, B1, B4, B7 as shown in the diagram below), by copying the contents of the SHARING runtime environment A1 batch parm deck.

6. In each new batch parm deck, review and modify the values accordingly.

7. Submit the Configuration tool batch job for each of the other LPARs. This creates the required runtime environments for the other LPARs.

8. Copy the RTE BASE datasets to the LPAR B1 (Target), using any copy/rename method or the supplied Configuration tool transport runtime environment batch jobs. Keep the same dataset names, if possible.

9. Copy and rename the appropriate runtime environment to each execution LPAR, using any copy/rename method or the supplied Configuration tool transport runtime environment batch jobs. As the files are copied, rename them to match the execution LPAR name.
10. Complete the configuration. At a minimum you must do the following:
   - Copy the started tasks and VTAM major nodes to the system libraries.
   - APF-authorize the datasets on the appropriate LPARs.

11. Start each of the component products.

Figure 16. Two Systems with 6 LPARs in Scenario 5

<table>
<thead>
<tr>
<th>CPC A</th>
<th>LPC A3</th>
<th>LPC A4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shared DASD within CPC A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 Configuration tool and 1 SMP/E environment (includes Target libraries)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 RTE BASE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 Unique RTEs</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LPC A1 (Main)</th>
<th>LPC A3</th>
<th>LPC A4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration tool, SMP/E, Target libraries</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RTE BASE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SHARING RTE A1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>{RTE A3}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>{RTE A4}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>{RTE B1}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>{RTE B4}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>{RTE B7}</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CPC B</th>
<th>LPC B1</th>
<th>LPC B4</th>
<th>LPC B7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shared DASD within CPC B</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 Copy of RTE BASE</td>
<td>&lt;RTE BASE&gt;</td>
<td>&lt;RTE B4&gt;</td>
<td>&lt;RTE B7&gt;</td>
</tr>
<tr>
<td>3 Copies of RTEs</td>
<td>&lt;RTE B1&gt;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Legend: <item> = Copied from main LPAR A1
{item} = Created with Configuration tool Batch
Scenario 6: Replicating a Runtime Environment Using Interactive Copy

This scenario describes a method for creating a second runtime environment that is based on values specified in another runtime environment.

Pros and Cons of using this scenario

Table 64. Pros and Cons of Scenario 6

<table>
<thead>
<tr>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Saves time by jump starting a new runtime environment with</td>
<td>Still need to navigate through all panels to review values and submit</td>
</tr>
<tr>
<td>configuration values already specified in an existing runtime</td>
<td>all configuration jobs</td>
</tr>
<tr>
<td>environment</td>
<td></td>
</tr>
</tbody>
</table>

Procedure

To use this scenario, perform the following steps:

1. Create and configure runtime environment A. For instructions on how to create a runtime environment, see “Setting up the Runtime Environment” on page 79.

2. Create runtime environment B by jump-starting it with all the configuration values you specified for runtime environment A. Follow the instructions in “Adding a Runtime Environment” on page 85. In the process, make sure to enter the name of runtime environment A in the “Copy configuration values from RTE” field, as described in the table “First Add Runtime Environment Panel” on page 85.

Note: You must complete all configuration steps and jobs for runtime environment B.
Scenario 7: Copying Configuration Values from One Configuration Tool to Another Configuration Tool

This scenario describes a method for copying configuration values (one runtime environment at a time), from one Configuration tool to another. This can also be thought of as exporting and importing.

Pros and Cons of using this scenario

Table 65. Pros and Cons of Scenario 7

<table>
<thead>
<tr>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Saves time because you do not need to enter configuration values into the new Configuration tool</td>
<td>None</td>
</tr>
</tbody>
</table>

Procedure

1. In Configuration tool #1, do the following:
   1. Create runtime environment A.
   2. Generate the Configuration tool batch parm deck for runtime environment A.
2. In Configuration tool #2, create the Configuration tool batch mode job CICATB.
3. Submit CICATB for Configuration tool #2, using the parm deck for runtime environment A.

   **Note:** To only populate the Configuration tool ISPF tables and create the runtime environment construction JCL in INSTJOBS, you can enter `SUBMIT(NO)`. The jobs will not be submitted to construct the runtime environment.

Figure 17. Copying Configuration Values from One Configuration Tool to Another

<table>
<thead>
<tr>
<th>Configuration Tool #1</th>
<th>Configuration Tool #2</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTE A</td>
<td>Run CICATB using batch parm deck from Configuration tool #1 RTE A.</td>
</tr>
<tr>
<td>Create batch parm deck for RTE A</td>
<td>All RTE values will be stored into ISPF tables of Configuration tool #2</td>
</tr>
</tbody>
</table>
Scenario 7: Copying Configuration Values from One Configuration Tool to Another Configuration Tool
Introduction

This appendix describes the Configuration tool’s configuration services and utilities.

You can use the configuration services and utilities to perform various services on the runtime environment and specify diagnostic information. Some of the services may modify the Configuration tool values stored in ISPF tables.

Note: You should not modify any values unless you are told to do so in the documentation or by IBM Software Support personnel. If the Configuration tool values are modified incorrectly, the Configuration tool could stop functioning or produce unpredictable results.

Appendix contents

Services: Unlocking and Modifying Runtime High-Level Qualifiers ......................... 212
Services: Creating the Configuration Tool Batch Mode Job ................................. 213
Utilities: Specifying DEBUG Options ............................................................... 214
Utilities: Displaying the Contents of an ISPF Table ............................................. 215
Utilities: Selecting and Executing a CLIST in the TKANCUS Library .................... 216
Services: Unlocking and Modifying Runtime High-Level Qualifiers

Using this option with care

You can use this option to unlock the high-level qualifier values that you specified when you were setting up the configuration environment. If you need to modify these values, you must first unlock them.

Warning: If you unlock and change the high-level qualifiers, the Configuration tool does not automatically delete and reallocate the existing libraries. The jobs generated by the Configuration tool will fail if they are pointing at incorrect libraries.

Procedure

Perform the following steps to unlock and modify runtime high-level qualifiers:

1. From the Main Menu, select Configure products > Services and utilities.
   The Configuration Services and Utilities panel displays.

2. Select Unlock runtime high-level qualifiers.
   The Unlock Runtime High-Level Qualifiers panel displays.

3. Enter Y in the “Unlock runtime high-level qualifiers?” field and press Enter.
   The Set Up Configuration Environment panel displays.

4. Modify the high-level qualifiers.
Services: Creating the Configuration Tool Batch Mode Job

You can use this option to generate the Job Control Language (JCL) that runs the
Configuration tool steps under batch.
This option also creates the KCISETUP REXX exec. Invoking KCISETUP enables the
ISPF environment to use the ISPF macros provided with the Configuration tool. You can
use these macros to compose and manage the parameter members used for the
Configuration tool batch mode process.

Procedure

Perform the following steps to create the Configuration tool batch mode job:

1. From the Main Menu, select Configure products > Services and utilities.
The Configuration Services and Utilities panel displays.

2. Select Create batch mode job.
The Configuration tool displays a message at the top of the panel indicating the job has
been created.

3. To view additional information about this job, press F1.
Utilities: Specifying DEBUG Options

Procedure

Perform the following steps to specify or modify DEBUG parameter values:

1. From the Main Menu, select **Configure products > Services and utilities**. The Configuration Services and Utilities panel displays.

2. Select **DEBUG options**. The DEBUG Options panel displays with all of the existing DEBUG values that you entered when invoking the Configuration tool.

3. Contact IBM Software Support. IBM Software Support personnel will direct you in specifying or modifying the DEBUG parameter values.
Utilities: Displaying the Contents of an ISPF Table

You can use this option to specify the contents of an ISPF table located in the data library.

Procedure

Perform the following steps to display an ISPF table:

1. From the Main Menu, select **Configure products > Services and utilities**.
   The Configuration Services and Utilities panel displays.

2. Select **Display an ISPF table**.
   The Display an ISPF Table panel displays.

3. Specify the name of the ISPF table you want to display.
   You can limit the information displayed for an ISPF table by specifying one to three sets of display criteria under “Optional section parameters”. For each set you must specify the variable name and matching value.

4. Press Enter to view the ISPF table you specified.
   The actions you can use on the table are the following:

<table>
<thead>
<tr>
<th>Key to Press</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>END (PF3)</td>
<td>Go to previous record</td>
</tr>
<tr>
<td>ENTER</td>
<td>Go to next record</td>
</tr>
<tr>
<td>CANCEL</td>
<td>Go back to prior panel</td>
</tr>
<tr>
<td>UP/DOWN</td>
<td>Use scroll variables</td>
</tr>
</tbody>
</table>
Utilities: Selecting and Executing a CLIST in the TKANCUS Library

You can use this option to execute a specific CLIST/REXX exec that is in the TKANCUS library.

Procedure

Perform the following steps to select and execute a specific CLIST in TKANCUS:

1. From the Main Menu, select Configure products > Services and utilities. The Configuration Services and Utilities panel displays.
2. Select Execute a CLIST in the TKANCUS library.
3. Contact IBM Software Support. IBM Software Support personnel will direct you in selecting and executing a CLIST in the TKANCUS library.
Introduction

This appendix describes the Configuration tool’s batch utilities.

Appendix contents

Using the Configuration Tool Batch Utilities ................................................. 218
KCISETUP: Setting Up the Environment ......................................................... 219
KCICFKEY: Managing PF Keys ................................................................. 220
KCICPGHP: Displaying Batch Parm Deck Member Parameters Help .................. 222
KCICPGEN: Generating the Batch Parameter Input Deck ............................ 224
Using the Configuration Tool Batch Utilities

The following are the four Configuration tool utilities that are available if you are using batch mode processing:

Table 67. Configuration Tool Batch Utilities

<table>
<thead>
<tr>
<th>Utility</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KCISETUP</td>
<td>Sets up the environment that is needed to use the Configuration tool batch utilities.</td>
</tr>
<tr>
<td>KCICFKEY</td>
<td>Manages the PF keys that are used for the Configuration tool batch utilities.</td>
</tr>
<tr>
<td>KCICPGHP</td>
<td>Displays online help information for parameters in a batch parameter deck member.</td>
</tr>
<tr>
<td>KCICPGEN</td>
<td>Adds component products to a batch parameter deck member.</td>
</tr>
</tbody>
</table>

These utilities are designed to run outside the Configuration tool, but can also be used while in the Configuration tool.
KCISETUP: Setting Up the Environment

You use the KCISETUP utility to set up the environment that is necessary for using the other Configuration tool batch utilities. This utility must be run once after starting the TSO ISPF session and can only be run from an ISPF session.

Before using the KCISETUP utility, you must generate the KCISETUP member in the INSTLIB. KCISETUP can only be run once per session. There is no confirmation message issued to indicate successful completion of KCISETUP.

Generating KCISETUP in the INSTLIB

Perform the following steps to generate KCISETUP:

1. Start the Configuration tool on the master image.
2. From the Main Menu, select Configure products > Services and utilities.
   The Configuration Services and Utilities panel displays.
3. Select Create batch mode job.
   The Configuration tool generates member KCISETUP in the INSTLIB.

   Note: KCISETUP must be created once on an image and can be used for all subsequent parameter deck processing on that image. If the ISPF environment changes or you split the INSTLIB, you must recreate KCISETUP.

Invoking the environment setup

You can invoke the environment setup utility by performing one of the following procedures:

- From an ISPF command line, enter the following:

  **TSO EXEC 'shilev.INSTLIB(KCISETUP)'**

  where the variable shilev is the SMP/E dataset high-level qualifier.

- From an ISPF Primary Option Menu panel do the following:
  A. Select Enter TSO or Workstation.
     The ISPF Command Shell panel displays.
  B. Enter the following:

     **EXEC 'shilev.INSTLIB(KCISETUP)'**

     where the variable shilev is the SMP/E dataset high-level qualifier.
You use the KCICFKEY utility to manage ISPF session PF keys that are used for batch utilities. This includes turning the PF keys on and off, and toggling which set of keys display. This utility can only be run under an ISPF session.

**Note:** If you will be using KCICFKEY to manage the ISPF session PF keys for the batch utilities, you must turn on the pre-defined function keys. To do this, issue the **PFSHOW** command from either the ISPF command line or any of the Configuration tool panel command lines.

### Before using KCICFKEY

Before using this Configuration tool batch utility, you must generate KCISETUP to set up the environment. See “Generating KCISETUP in the INSTLIB” on page 219.

### Setting up the ISPF session to support 24 PF keys

To use the KCICFKEY utility, the ISPF session must support 24 PF keys. Perform the following steps to set up the ISPF session to support 24 PF keys:

1. From the ISPF Primary Option Menu, select **Terminal and user parameters**
   
   The ISPF Settings panel displays.

2. From the Function keys menu, select **Non-Keylist PF Key settings**.
   
   The PF Key Definitions and Labels panel displays.

3. Enter **24** for Number of PF Keys.

4. Press F3 to return to the ISPF Primary Option Menu.
   
   If the ISPF session is not setup to support 24 PF keys, the KCICFKEY utility will run but will issue the following ISPF dialogue warning message:

   **“PFKEYS COUNT ERROR”, “Number of PF Keys must be 24.  See ISPF Settings.”**

### Features of the Configuration tool PF Key Manager

When setting PF keys, the Configuration tool PF Key Manager will own PF keys 13–24. On keyboards that do not support 24 PF keys, PF keys 13–24 are obtained by holding the Shift key and pressing a function key. While the shift key is pressed, function keys 1–12 become 13–24.

When the Configuration tool PF keys are active, any change in the PF Key Show State is preserved. If you have set the PF Key Show State to Show All and then turned the PF keys off, when you turn the PF keys back on, the PF Key Show State is restored to Show All.

While using the Configuration tool PF Key Manager, all of the original PF key and Show State settings are preserved. After exiting the PF Key Manager, all of the original PF key and Show State settings are restored.
**Invoking the PF Key Manager**

You invoke the PF Key Manager utility by performing one of the following procedures:

- From an ISPF command line, enter the following:
  
  ```
  TSO KCICFKEY state
  ```

- From an ISPF Primary Option Menu panel do the following:
  
  A. Select **Edit**.
     
     The Edit Entry panel displays.
  
  B. Enter the following:
     
     ```
     KCICFKEY state
     ```

     where the variable `state` is the desired state of the Configuration tool PF keys.

     Valid states are the following:

     **Table 68. PF Keys - Valid States**

     | State | Description |
     |-------|-------------|
     | ON    | Turn on the PF keys. |
     | OFF   | Turn off the PF keys. |
     | SHOW  | If the PF keys are active, then toggle the PF keys between All (1–24), Alternate (13–24), and Primary (1–12). |
     | HELP  | Display the PF Key Manager help information. |

     If `state` is not specified, the Configuration tool PF keys will toggle between ON and OFF.

**PF keys for batch utilities**

The Configuration tool PF Key Manager sets the following keys for the batch utilities:

**Table 69. PF Keys for Batch Utilities**

<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PF13</td>
<td>Displays the PF Key Manager Help information</td>
</tr>
<tr>
<td>PF14</td>
<td>Provides detail help information for the Batch Parameter, on the line where the cursor is positioned</td>
</tr>
<tr>
<td>PF15</td>
<td>Turns the PF keys off</td>
</tr>
<tr>
<td>PF21</td>
<td>Toggles the PF keys shown at the bottom of the display between All (1–24), Alternate (13–24), and Primary (1–12)</td>
</tr>
</tbody>
</table>
You use the KCICPGHP utility to display help information for parameters in a batch parameter deck member. The detailed help information for each of the batch parameters is provided to help you modify or construct a batch parameter deck.

This utility must be run from an ISPF Edit session.

Before using KCICPGHP

Before using this batch utility, you must generate KCISETUP to set up the environment. See “Generating KCISETUP in the INSTLIB” on page 219.

Invoking the Batch Parameter Deck Help utility

Note: To use this utility you must be running under an ISPF session, editing a member or dataset.

You invoke the Batch Parameter Deck Help utility by performing one of the following procedures:

1. From an ISPF Primary Option Menu panel, select Edit.
The Edit Entry panel displays.

2. Enter the following:
   KCICPGHP

3. Position the cursor on the row that contains a batch parameter, and then perform one of the following procedures:
   - Press Enter.
   - Press the PF key assigned by the Configuration tool PF Key Manager.

Note: The latter is the preferred method for invoking Batch Parameter Deck Help. The PF Key Manager will assign a PF key to invoke this function. With either method, you must position the cursor on the row that contains the batch parameter. The utility will then isolate the parameter, perform a look-up, and display a pop-up dialogue with the detailed help information.

Format of the batch parameter helps

The batch parameter helps contain the following four sections:

Table 70. Batch Parameter Format

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title Information</td>
<td>The parameter name, followed by a short description of the parameter</td>
</tr>
<tr>
<td>Description Area</td>
<td>The detailed help information for the parameter. This area is scrollable, as indicated by the (+) indicator on the bottom right.</td>
</tr>
<tr>
<td></td>
<td>PF7 and PF8 are assigned to scroll this area.</td>
</tr>
</tbody>
</table>
Sample batch parameter help

The following is an example of a batch parameter help:

**Figure 18. Example of a Batch Parameter Help**

<table>
<thead>
<tr>
<th>KMV_CMS_NAME</th>
<th>CMS Name</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description:</strong></td>
<td>This is the nodeid of the CMS to which you are connecting the agent.</td>
</tr>
<tr>
<td></td>
<td>This name must match the domain name of a non-z/OS CMS, or the nodeid parameter in the KDSCNFG member of the RKANPAR library for a z/OS CMS. If the NODEID parameter contains the literal *SMFID, the CMS Name definition must use the actual z/OS SMFID in place of this literal value.</td>
</tr>
<tr>
<td></td>
<td>The value of this field is case sensitive for both z/OS and</td>
</tr>
<tr>
<td><strong>Attributes:</strong></td>
<td>(+)</td>
</tr>
<tr>
<td>Required:</td>
<td>Yes</td>
</tr>
<tr>
<td>Maximum Length:</td>
<td>32</td>
</tr>
<tr>
<td>Type of Data:</td>
<td>Character (Mixed Case)</td>
</tr>
<tr>
<td>Default value:</td>
<td></td>
</tr>
<tr>
<td>F1 = Help  F3 = End  F5 = Show All  ** = Backward  F8 = Forward</td>
<td></td>
</tr>
</tbody>
</table>
You use the KCICPGEN utility to add component products to a batch parameter deck member or to construct a batch parameter deck from scratch.

Note: For a component product that has already been installed, KCICPGEN does not support the addition of a new component product release.

Before using KCICPGEN

Before using this Configuration tool batch utility, you must generate KCISETUP to set up the environment. See “Generating KCISETUP in the INSTLIB” on page 219.

Invoking KCICPGEN

Note: To use this utility you must be running under an ISPF session, editing a member or dataset.

You invoke the Batch Parameter Input Deck Generator by performing the following procedure:

1. From an ISPF Primary Option Menu panel, select Edit.
   The Edit Entry panel displays.
2. Enter the following:
   
   KCICPGEN product
   
   where the variable product is the component product on which you will perform the operation.

The format of this parameter can be either of the following:

Table 71. KCICPGEN Format

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kppvvv</td>
<td>For a specific component product, the following are the variables: pp = an alpha-numeric product code of length two vvv = a numeric version number of length three</td>
</tr>
<tr>
<td>BASERTE</td>
<td>For a base runtime environment build</td>
</tr>
</tbody>
</table>

Note: When adding a component product, only supply the component code and version that identifies the package that you purchased. For example, if you are adding a Plex component product such as OMEGAMON XE for Sysplex V220, enter KOS220. The underlying component products that make up the package, such as OMEGAMON II for MVS, will be included.

The KCICPGEN utility invokes the Configuration tool edit macro. The edit macro uses the Kppvvv or QI parameter to locate the correct component product information (PI) member within INSTDATA. Once the PI member has been found, the edit macro scans the file and builds a list of all the components and parameters required for the specified
component product. If a component product already exists within the library member, it will not be replaced.
KCICPGEN: Generating the Batch Parameter Input Deck
If you have a problem with your IBM software, you want to resolve it quickly. This section describes the following options for obtaining support for IBM software products:

- “Searching knowledge bases” on page 227
- “Obtaining fixes” on page 228
- “Receiving weekly support updates” on page 228
- “Contacting IBM Software Support” on page 229

**Searching knowledge bases**

You can search the available knowledge bases to determine whether your problem was already encountered and is already documented.

**Searching the information center**

IBM provides extensive documentation that can be installed on your local computer or on an intranet server. You can use the search function of this information center to query conceptual information, instructions for completing tasks, and reference information.

**Searching the Internet**

If you cannot find an answer to your question in the information center, search the Internet for the latest, most complete information that might help you resolve your problem.

To search multiple Internet resources for your product, use the Web search topic in your information center. In the navigation frame, click Troubleshooting and support > Searching knowledge bases and select Web search. From this topic, you can search a variety of resources, including the following:

- IBM technotes
- IBM downloads
- IBM Redbooks®
- IBM developerWorks®
- Forums and newsgroups
- Google
Obtaining fixes

A product fix might be available to resolve your problem. To determine what fixes are available for your IBM software product, follow these steps:

2. Click Downloads and drivers in the Support topics section.
3. Select the Software category.
4. Select a product in the Sub-category list.
5. In the Find downloads and drivers by product section, select one software category from the Category list.
6. Select one product from the Sub-category list.
7. Type more search terms in the Search within results if you want to refine your search.
8. Click Search.
9. From the list of downloads returned by your search, click the name of a fix to read the description of the fix and to optionally download the fix.

For more information about the types of fixes that are available, IBM Software Support Handbook at http://techsupport.services.ibm.com/guides/handbook.html.

Receiving weekly support updates

To receive weekly e-mail notifications about fixes and other software support news, follow these steps:

2. Click My Support in the upper right corner of the page.
3. If you have already registered for My Support, sign in and skip to the next step. If you have not registered, click register now. Complete the registration form using your e-mail address as your IBM ID and click Submit.
4. Click Edit Profile.
5. In the Products list, select Software. A second list is displayed.
6. In the second list, select a product segment, for example, Application servers. A third list is displayed.
7. In the third list, select a product sub-segment, for example, Distributed Application & Web Servers. A list of applicable products is displayed.
8. Select the products for which you want to receive updates, for example, IBM HTTP Server and WebSphere Application Server.
9. Click Add products.
10. After selecting all products that are of interest to you, click Subscribe to email on the Edit profile tab.
11. Select Please send these documents by weekly email.
12. Update your e-mail address as needed.

13. In the **Documents** list, select **Software**.

14. Select the types of documents that you want to receive information about.

15. Click **Update**.

If you experience problems with the **My support** feature, you can obtain help in one of the following ways:

**Online:** Send an e-mail message to erchelp@ca.ibm.com, describing your problem.

**By phone:** Call 1-800-IBM-4You (1-800-426-4968).

### Contacting IBM Software Support

IBM Software Support provides assistance with product defects.

Before contacting IBM Software Support, your company must have an active IBM software maintenance contract, and you must be authorized to submit problems to IBM. The type of software maintenance contract that you need depends on the type of product you have:

- For IBM distributed software products (including, but not limited to, Tivoli, Lotus®, and Rational® products, as well as DB2® and WebSphere® products that run on Windows or UNIX operating systems), enroll in Passport Advantage® in one of the following ways:
  - **Online**: Go to the Passport Advantage Web page (http://www.lotus.com/services/passport.nsf/WebDocs/Passport_Advantage_Home) and click **How to Enroll**
  - **By phone**: For the phone number to call in your country, go to the IBM Software Support Web site at http://techsupport.services.ibm.com/guides/contacts.html and click the name of your geographic region.

- For customers with Subscription and Support (S & S) contracts, go to the Software Service Request Web site at https://techsupport.services.ibm.com/ssr/login.


- For IBM eServer™ software products (including, but not limited to, DB2 and WebSphere products that run in zSeries, pSeries, and iSeries environments), you can purchase a software maintenance agreement by working directly with an IBM sales representative or an IBM Business Partner. For more information about support for eServer software products, go to the IBM Technical Support Advantage Web site at http://www.ibm.com/servers/eserver/techsupport.html.

If you are not sure what type of software maintenance contract you need, call 1-800-IBMSERV (1-800-426-7378) in the United States. From other countries, go to the contacts page of the **IBM Software Support Handbook** on the Web at
To contact IBM Software Support, follow these steps:

1. “Determining the business impact” on page 230
2. “Describing problems and gathering information” on page 230
3. “Submitting problems” on page 231

### Determining the business impact

When you report a problem to IBM, you are asked to supply a severity level. Therefore, you need to understand and assess the business impact of the problem that you are reporting. Use the following criteria:

<table>
<thead>
<tr>
<th>Severity 1</th>
<th>The problem has a <em>critical</em> business impact. You are unable to use the program, resulting in a critical impact on operations. This condition requires an immediate solution.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Severity 2</td>
<td>The problem has a <em>significant</em> business impact. The program is usable, but it is severely limited.</td>
</tr>
<tr>
<td>Severity 3</td>
<td>The problem has <em>some</em> business impact. The program is usable, but less significant features (not critical to operations) are unavailable.</td>
</tr>
<tr>
<td>Severity 4</td>
<td>The problem has <em>minimal</em> business impact. The problem causes little impact on operations, or a reasonable circumvention to the problem was implemented.</td>
</tr>
</tbody>
</table>

### Describing problems and gathering information

When explaining a problem to IBM, be as specific as possible. Include all relevant background information so that IBM Software Support specialists can help you solve the problem efficiently. To save time, know the answers to these questions:

- What software versions were you running when the problem occurred?
- Do you have logs, traces, and messages that are related to the problem symptoms? IBM Software Support is likely to ask for this information.
- Can you re-create the problem? If so, what steps were performed to re-create the problem?
- Did you make any changes to the system? For example, did you make changes to the hardware, operating system, networking software, and so on.
- Are you currently using a workaround for the problem? If so, be prepared to explain the workaround when you report the problem.
- What software versions were you running when the problem occurred?
Submitting problems

You can submit your problem to IBM Software Support in one of two ways:

- **Online**: Click **Submit and track problems** on the IBM Software Support site at [http://www.ibm.com/software/support/probsub.html](http://www.ibm.com/software/support/probsub.html). Type your information into the appropriate problem submission form.

- **By phone**: For the phone number to call in your country, go to the contacts page of the IBM Software Support Handbook ([http://techsupport.services.ibm.com/guides/contacts.html](http://techsupport.services.ibm.com/guides/contacts.html)) and click the name of your geographic region.

If the problem you submit is for a software defect or for missing or inaccurate documentation, IBM Software Support creates an Authorized Program Analysis Report (APAR). The APAR describes the problem in detail. Whenever possible, IBM Software Support provides a workaround that you can implement until the APAR is resolved and a fix is delivered. IBM publishes resolved APARs on the Software Support Web site daily, so that other users who experience the same problem can benefit from the same resolution.
Overview

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

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

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

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

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

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

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

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

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

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

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

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

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

COPYRIGHT LICENSE:

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

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

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

Trademarks

IBM, the IBM logo, AIX, Candle, Candle Management Server, Candle Management Workstation, CandleNet, CandleNet Portal, CICS, DB2, developerWorks, eServer, IBMLink, Informix, iSeries, Lotus, OMEGAMON, Passport Advantage, pSeries, Rational, Redbooks, S/390, Tivoli, the Tivoli logo, Tivoli Enterprise Console, TME, VTAM, RACF, WebSphere, z/OS, and zSeries are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

Intel, Intel Inside (logos), MMX, Celeron, Intel Centrino, Intel Xeon, Itanium, Pentium and Pentium III Xeon are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States, other countries, or both.

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

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

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.
access control levels 152
access control list
  see ACL
accessing
  Configuration tool 76, 77
  queue managers 42
  your queue managers 146
ACF2 87
ACL
  how the ACL affects authorization of IBM Tivoli
  OMEGAMON XE for WebSphere Integration
  Brokers 162
adding
  new component products to an existing runtime
  environment 80
  NLS component products to an existing runtime
  environment 81
  runtime environment 80
  new 85
address translation 60
advanced features 46
Advanced key 50
agent address space 63
  advanced configuration 114
  agent running in 36, 39
  specifying 110
  communication values 112
agents 54, 56
  connecting to CMS 111
  connecting to secondary CMS 114
  platforms 57
  types 56
alert adapters 56
alert emitters 56
allocating
  PDSE datasets 65
  runtime libraries 89
APF-authorizing load libraries 41, 43
applid prefix, global 87
authorization-checking levels 150
authorized load library 32, 35
authorizing external security program 42, 146
authorizing IBM Tivoli OMEGAMON XE for
WebSphere Integration Brokers 43, 161
  additional 162
base libraries 64, 65
BASE runtime environment 64, 69
Batch Mode
  parameter decks 181
  batch mode
    overview 181
    parameter decks 181, 185
    runtime environment parameter export 181, 184
    runtime environment transport 182, 187
  batch mode processing 46, 51, 65, 180
Batch Parameter Deck Help utility 222
batch utilities 217
  KCICFKEY 220
  KCICPGEN 224
  KCICPGHP 222
  KCISETP 219
books 16
  see publications 16
browser mode 56
building
  runtime environment libraries 80
  runtime libraries 89
Candle Management Server
  see CMS
Candle Management Workstation 54
CandleMonitor Node
  installing 163
    permanently 163
    temporarily 164
CandleNet Portal 54, 55
  component products 56
CandleNet Portal browser client 56
CandleNet Portal desktop client 55, 56
CandleNet Portal Server 54, 56
CA-TOP SECRET
  authorizing monitoring agent using 147
  example 147
changing the configuration 45, 168
checklists 21, 22
CICS channels 149
CMS 54
changing connection 114
configuring on a Windows machine 131
hub 58
local 58
installing agent into 118
registering with 94
name of instance of 86
non-local 58
non-z/OS, connecting to 115
platforms 54, 57
remote 57, 58
secondary CMS specification for agents 114
seeding on a remote machine 41, 43, 137
seeding on a UNIX machine 134, 136
seeding on a Windows machine 132
types 58
CMW
see Candle Management Workstation
collect enhanced audit log information 32
communication protocol requirements 60
communication protocol values
specifying for agent address spaces 112
completing the configuration 34, 37, 39
component product upgrade alert 66
component products 48
CandleNet Portal 56
installing prerequisite OMEGAMON Platform and
CandleNet Portal 60
OMEGAMON XE 54
configuration
considerations 66
for DASD sharing environments 70, 71
for limited DASD space 72
select a component product 83
services and utilities 83
configuration database
changing type to DB2 96
choosing type 94
migrating 95
migrating to DB2 95
securing 150
setting up on a UNIX machine 41, 138
setting up on a Windows machine 131
configuration parameters
creating 123
specifying 102
configuration scenarios
copying configuration values from one
Configuration tool to another Configuration
tool 209
location of supporting topics 194
replicating a runtime environment
on same LPAR using batch processing 206
to another LPAR using batch Auto-submit 203
to another LPAR using batch processing 201
using Interactive Copy 208
rolling out component products
across a 40 LPAR Enterprise (Common
RKANPAR and Common RKANCMD
Libraries) 198
across a 40 LPAR Enterprise (Standard Method)
195
Configuration tool 48
accessing 76, 77
batch components 180
batch mode processing 179, 180, 181
commands and functions 49, 50
defaults 49
displaying information in 50
messages 49
online help 49
restrictions 49
setting up environment for 76
starting 76, 77
configured system 151
configured system group 151
configuring
agents in their own agent address space 63
checklist for updating or changing 45
CMS on a Windows machine 131
completing outside the Configuration tool 125, 145
component products
in an existing runtime environment 28
definition 48
IBM Tivoli OMEGAMON XE for WebSphere
Integration Brokers 38
IBM Tivoli OMEGAMON XE for WebSphere MQ
Configuration 31
IBM Tivoli OMEGAMON XE for WebSphere MQ
Monitoring 35
in a runtime environment 25, 80
existing 25
new 25
outside Configuration tool 41, 43
Persistent Data Store 31, 37, 39, 41, 43, 97
prerequisites 23
queue managers 146
VTAM applid considerations 66
with remote CMS 59
copying 129
started task procedures 41, 43
VTAM definitions 41, 43
creating
configuration parameters 38, 123
new runtime environment 67, 80
for NLS component products 81
runtime environment for another z/OS image 65
runtime members 33, 36, 39, 119
scheduled actions 157
single IBM Tivoli VTAM major node 177
system variable parameter member 175
VTAM major node rename job 176
CT/PDS
see Persistent Data Store
cumulative maintenance 48
customer support
see Software Support 229
customizing
agent parameter files for IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers 165
definition 48

D
DB2 configuration database 94, 131, 138, 139
default monitoring group 35
defaults provided with Configuration tool 49
defined configurations 153
deleting
runtime environment 92
scheduled actions 157
description of package and component products 13
description of this guide 13
desktop mode 55, 56
different address space 42, 44
displaying information in the Configuration tool 50
documentation CD 16
documentation conventions 19

easiest runtime environment to create 68
edit recovery 49
editing global variables 157
enabling
historical data collection 37
system variable support 171, 173
End key 50
enhanced audit log 157
Enter key 50
examples of generic profiles 158
existing CSI

copying the installation library 76
external security program 146

F
FULL runtime environment 64, 68
functions of OMEGAMON XE 54

gateways 57
generating a batch parameter deck 224
generic profiles 157
global
applid prefix 87
DASD information 82
started task prefix 86
VTAM major node 127
granting authorizations to agents 146

H
HELP command 50
high-level qualifiers
non-VSAM datasets 82
non-VSAM libraries 86
VSAM datasets 82
VSAM libraries 86
HPNS TCP/IP protocol support 61
hub CMS 58

I
IBM Tivoli OMEGAMON XE for WebSphere Business Integration 16
IBM Tivoli OMEGAMON XE for WebSphere Integration Brokers
agent parameter files, customizing 165
authorizing 43, 161
additional 162
effects of ACL 162
installing CandleMonitor Node 163
permanently 163
temporarily 164
IBM Tivoli OMEGAMON XE for WebSphere MQ Configuration
migrating elements from previous version 34
implementing security
configuration database 159
from CandleNet Portal 160
from z/OS 159
multi-level 42
information centers, searching to find software
problem resolution 227
installing
agent into local CMS 33, 37, 118
definition 48
prerequisite OMEGAMON Platform and
CandleNet Portal component products 60
product support on a non-z/OS machine 41, 43,
130
INSTJOBS 176
interactive mode 51
Internal configuration database 94
ISPF display requirements 49
J
JES3 support 66
K
KCICFKEY 218, 220
KCICPGEN 218, 224
KCICPGHP 218, 222
KCISSETUP 218, 219
knowledge bases, searching to find software problem
resolution 227
L
language library 32, 35
libraries
LPAR-specific 64, 65
linkedit libraries
specifying 38, 122
load libraries 128
loading
libraries
runtime 91
runtime environment 80
local CMS 42, 44, 58
M
managing
ISPF session PF keys 220
your environment 83
manuscripts 16
see publications 16
messages in Configuration tool 49
mid-level qualifier 85
migrating 48
elements from previous version of IBM Tivoli
OMEGAMON XE for WebSphere MQ
Configuration 34
migration utility
running 120
modifying scheduled actions 157
monitoring agents 56
monitoring CICS channels 42, 149
multi-level security 150, 159
N
National Language Support
see NLS
network ID 60
network interface card 60
new CSI
creating the installation library 77
setting up your work and configuration
environments 77
NLS
runtime environment configuration sequence 81
non-local CMS 42, 44, 58
non-z/OS CMS
connecting to 115
O
OMEGAMON Platform and CandleNet Portal 16
prerequisite installation 60
OMEGAMON XE
component products 54
functions 54
overview 54
online help 49
online publications
accessing 16
operating systems
agents 57
CMS 54, 57
ordering publications 17
P
partition name 60
PDSE support 65
Persistent Data Store 37
configuring 97
copying procedures 129
Index 241

creating or editing maintenance jobcard 99
PF Key Manager 220
PFSHOW command 49
planning
  your configuration 53
  your runtime environment 67
platforms
  agents 57
  CMS 54, 57
possible configurations using runtime environments 68
prefix
  applid 87
  started tasks 86
preparing queue managers for monitoring 62
preventive maintenance
  definition 48
procedures
  copying for the Persistent Data Store 129
  copying for the started tasks 126
processing
  of applids by the Configuration tool 66
PROCLIB
  Persistent Data Store 129
  started tasks 126
product support
  installing on a non-z/OS machine 41, 43, 130
  selecting on a UNIX machine 134, 136
  selecting on a Windows machine 130
profile types 153
profile-naming conventions 153
profiles you can create 152
prototype configurations 156
publications 16
  accessing online 16
  ordering 17
Q
queue managers
  preparing for monitoring 62
R
RACF
  authorizing monitoring agent using 146
    class 159
    example 146
    profiles 150, 152, 157
README APP command 50, 66
README command 50
README ERR command 50
RECOVERY SMS command 65
RECOVERY SYS command 50
RECOVERY OFF command 49
RECOVERY ON command 49
reducing amount of disk space 65
registering with local CMS 31, 35, 38, 94
remote CMS 57, 58
  example configuration that includes 59
replicating
  products 42
  runtime environment 181
resource 152
resource group 151
restrictions on naming runtime environment
  high-level qualifiers 49
reviewing
  communication protocol requirements 60
  configuration database security 150
RKANPAR library 175
RKANSAM library 177
running migration utility 120
running software on any z/OS system 170
runtime environment 48
  adding 80
  new component products to 80
base libraries 64
building libraries 80, 89
configuration sequence 80
configuring component products 80
copying values 87
definition 64
exporting parameters 181
load optimization 66
loading libraries 80
private libraries 65
replication process 181
security 87
sharing 70, 71, 72
  with SMP/E 128
terms 64
transport methods 182
runtime environment high-level qualifiers
restrictions on naming 49
runtime libraries 64
runtime load libraries 128
runtime members
  creating 33, 36, 39, 119
SAF 150, 157
same address space 42
scenarios
  recommended configuration 193
secondary CMS feature 63
secondary CMS specification for agents 114
security 87
security switch profiles settings 147
seeding the CMS
  background 137
selecting
  component products to configure 83
product support on a UNIX machine 134, 136
product support on a Windows machine 130
services and utilities
  configuration 83
setting up
  component trace 62
configuration database on a UNIX machine 41, 138
configuration database on a Windows machine 131
configuration environment 24, 76
runtime environment configuration environment 82
shareable base libraries 69
sharing
  agent address space 63, 110
SHARING BASE 64
SHARING FULL 64
SHARING SMP/E TARGET 64
SMP/E
  target libraries 64
SMS support 65
SNMP gateways 57
Software Support
  contacting 229
specifying
  agent address space parameters 33, 36, 39, 110
configuration parameters 32, 38, 102
linkedit libraries 38, 122
monitoring parameters 35
runtime environment values 85
started tasks
  considerations for naming 66
  copying procedures 126, 128
  names 66
  prefix 86
starting
Configuration tool 76, 77
support for products
  installing on a non-z/OS machine 41, 43, 130
  selecting on a UNIX machine 134, 136
  selecting on a Windows machine 130
synchronizing maintenance across multiple systems 69
System Modification Program/Extended see SMP/E
system variable support 46, 65, 170
  existing runtime environments 173
  new runtime environments 173
system-specific parameter values 170
T
target load libraries 128
TCP/IP requirements 60
terminology 48
thin client 56
Tivoli software information center 16
turning on system variable support 173
types of runtime environment configurations 65, 68
U
updating
  configuration 45, 168
  site-specific information 82
using
  checklists in this guide 22
  existing runtime environment 67
  IPPipe 60
  IUCV interface 61
  multi-level security feature 150
  single VTAM major node 177
  UTIL command 50
utilizing system variable support 170
V
varying the major node active 41, 43
verifying the configuration 42, 44
  CMS on a different platform 143
  different address space 142
  same address space 141
viewing
  configuration information 83
  VTAM
  applid considerations 66
  applids 60
  major nodes on remote systems 176
VTAMLST
   copying the VTAM definition 127

W
WebSphere MQ default objects 62

Z
z/OS system symbolics 65