IBM Tivoli Composite Application Manager Agent for J2EE
Version 7.1.1

Installation and configuration guide

SC27-6228-00

IBM Tivoli Composite Application Manager Agent for J2EE
Version 7.1.1

Installation and configuration guide

SC27-6228-00

Note

Before using this information and the product it supports, read the information in “Notices” on page 109.

Contents

Figures

About this publication

This publication provides information about installing, customizing, starting, and maintaining IBM® Tivoli® Composite Application Manager Agent for J2EE on Windows, Linux, and UNIX systems.

Intended audience

This publication is for administrators or advanced users wanting to install or modify the configuration of ITCAM Agent for J2EE. The publication assumes that readers are familiar with maintaining operating systems, administering web servers, maintaining databases, and general information technology (IT) procedures. Specifically, readers of this publication must have some knowledge of the following topics:

- Operating systems on which you intend to install product components
- Java™ application servers, such as WebLogic, NetWeaver, JBoss, and Tomcat, and J2SE applications
- Internet protocols such as HTTP, HTTPS, TCP/IP, Secure Sockets Layer (SSL), and Transport Layer Security (TLS)
- Digital certificates for secure communication

Publications

This section lists publications in the product library and related documents. It also describes how to access Tivoli publications online and how to order Tivoli publications.

ITCAM for Applications library for Agents for WebSphere Applications, J2EE, and HTTP Servers

The following publications are included in the ITCAM for Applications library:

- *IBM Tivoli Composite Application Manager: Agents for WebSphere Applications, J2EE, and HTTP Servers User's Guide*

  Provides the user overview, user scenarios, and Helps for agents for WebSphere® Applications, J2EE, and HTTP Servers.
- *IBM Tivoli Composite Application Manager: Agents for WebSphere Applications, J2EE, and HTTP Servers Planning an Installation*

  Provides the user with a first reference point for installation or upgrade of agents for WebSphere Applications, J2EE, and HTTP Servers.
- *IBM Tivoli Composite Application Manager: Agent for WebSphere Applications Installation and Configuration Guide*

  Provides installation instructions for setting up and configuring ITCAM Agent for WebSphere Applications on distributed systems.
- *ITCAM Agent for J2EE Installation and Configuration Guide*

  Provides installation instructions for setting up and configuring ITCAM Agent for J2EE.
- *IBM Tivoli Composite Application Manager: Agent for HTTP Servers Installation and Configuration Guide*

Provides installation instructions for setting up and configuring ITCAM Agent for HTTP Servers.

- *IBM Tivoli Composite Application Manager: Agents for WebSphere Applications, J2EE, and HTTP Servers Troubleshooting Guide*

  Provides instructions on problem determination and troubleshooting for agents for WebSphere Applications, J2EE, and HTTP Servers.
- *IBM Tivoli Composite Application Manager: Agents for WebSphere Applications, J2EE, and HTTP Servers: Messaging Guide*

  Provides information about system messages received when installing and using agents for WebSphere Applications, J2EE, and HTTP Servers.
- *IBM Tivoli Composite Application Manager: Agent for WebSphere Applications Reporting Guide*

  Provides information about installing Agent for WebSphere Applications Reports and creating pre-defined and ad-hoc reports.

Related publications

The following documentation also provides useful information:

- IBM Tivoli Documentation Central:

  Information about IBM Tivoli Documentation is provided on the following website:

  https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Tivoli Documentation Central

Accessing terminology online

The IBM Terminology website consolidates the terminology from IBM product libraries in one convenient location. You can access the Terminology website at http://www.ibm.com/software/globalization/terminology .

Accessing publications online

The documentation CD contains the publications that are in the product library. The format of the publications is PDF, HTML, or both.

IBM posts publications for this and all other Tivoli products, as they become available and whenever they are updated, to the Tivoli Documentation Central website at https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Tivoli Documentation Central

Tip: 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: http://www.ibm.com/e-business/weblink/publications/servlet/pbi.wss.

You can also order by telephone by calling one of these numbers:

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

In other countries, contact your software account representative to order Tivoli publications. To locate the telephone number of your local representative, perform the following steps:

1. Go to http://www.ibm.com/e-business/weblink/publications/servlet/pbi.wss
2. Select your country from the list and click **Go**.
3. Click **About this site** in the main panel to see an information page that includes the telephone number of your local representative.

Accessibility

Accessibility features help users with a physical disability, such as restricted mobility or limited vision, to use software products successfully. With this product, you can use assistive technologies to hear and navigate the interface. You can also use the keyboard instead of the mouse to operate all features of the graphical user interface.

For additional information, see Appendix G, "Accessibility," on page 105.

Application Performance Management community on Service Management Connect

Connect, learn, and share with Service Management professionals: product support technical experts who provide their perspectives and expertise.

Access Service Management Connect at https://www.ibm.com/developerworks/servicemanagement/apm/index.html. Use Service Management Connect in the following ways:

- Become involved with transparent development, an ongoing, open engagement between other users and IBM developers of Tivoli products. You can access early designs, sprint demonstrations, product roadmaps, and prerelease code.
- Connect one-on-one with the experts to collaborate and network about Tivoli and the (enter your community name here) community.
- Read blogs to benefit from the expertise and experience of others.
- Use wikis and forums to collaborate with the broader user community.

Tivoli technical training

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

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

Tivoli user groups

Tivoli user groups are independent, user-run membership organizations that provide Tivoli users with information to assist them in the implementation of Tivoli Software solutions. Through these groups, members can share information and learn from the knowledge and experience of other Tivoli users. For more information about Tivoli Users Group, see www.tivoli-ug.org.

Support information

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

Online
Access the IBM Software Support site at http://www.ibm.com/software/support/probsub.html .

Troubleshooting Guide
For more information about resolving problems, see the *IBM Tivoli Composite Application Manager: Agents for WebSphere Applications, J2EE, and HTTP Servers Troubleshooting Guide*.

Conventions used in this publication

This publication uses several conventions for special terms and actions, operating system-dependent commands and paths, and margin graphics.

Typeface conventions

This publication uses the following typeface conventions:

Bold

- Lowercase commands and mixed case commands that are otherwise difficult to distinguish from surrounding text
- Interface controls (check boxes, push buttons, radio buttons, spin buttons, fields, folders, icons, list boxes, items inside list boxes, multicolumn lists, containers, menu choices, menu names, tabs, property sheets), labels (such as **Tip:**, and **Operating system considerations**:)
- Keywords and parameters in text

Italic

- Citations (examples: titles of publications, diskettes, and CDs
- Words defined in text (example: a nonswitched line is called a *point-to-point line*)
- Emphasis of words and letters (words as words example: "Use the word *that* to introduce a restrictive clause."; letters as letters example: "The LUN address must start with the letter *L*.")
- New terms in text (except in a definition list): a *view* is a frame in a workspace that contains data.
- Variables and values you must provide: ... where *myname* represents....

`Monospace`

- Examples and code examples
- File names, programming keywords, and other elements that are difficult to distinguish from surrounding text
- Message text and prompts addressed to the user
- Text that the user must type
- Values for arguments or command options

Operating system-dependent variables and paths

This guide refers to the following variables:

- *ITM_home*: the top-level directory for installation of IBM Tivoli Monitoring components, including ITCAM Agent for J2EE. The default location is `C:\IBM\ITM` on Windows systems and `/opt/IBM/ITM` on Linux and UNIX systems:
- *DC_home*: the directory where the data collector files are installed. The default location is:
  - On Windows systems, *ITM_home*`\j2eedc\`*DC_version*
  - On Linux and UNIX systems, *ITM_home*`/`*architecture*`/yj/j2eedc/`*DC_version*

Chapter 1. Overview of IBM Tivoli Composite Application Manager Agent for J2EE

Use ITCAM Agent for J2EE to monitor J2EE application servers and J2SE applications.

The current version of ITCAM Agent for J2EE supports the following types of application servers:

- WebLogic
- JBoss
- NetWeaver
- Tomcat

It also supports J2SE applications.

Overview of the monitoring and diagnostic capabilities

ITCAM Agent for J2EE can monitor J2EE application servers and J2SE applications, providing information within two different infrastructures: IBM Tivoli Monitoring and ITCAM for Application Diagnostics Managing Server.

The IBM Tivoli Monitoring environment places this agent into the context of the IBM Tivoli Monitoring family, a suite of products used to monitor a mixed-systems environment. With IBM Tivoli Monitoring, the user can:

- Monitor for alerts on the managed systems
- Trace the causes leading up to an alert
- Monitor processing time for various requests within J2EE and J2SE applications
- Establish your own performance thresholds
- Create custom situations, which are conditions that IBM Tivoli Monitoring automatically monitors
- Create and send commands to control system monitoring using the Take Action feature
- Create comprehensive reports about system conditions
- Define your own queries, using the attributes provided with ITCAM Agent for J2EE, to monitor conditions of particular interest to you

The Tivoli Enterprise Portal is the user interface for the IBM Tivoli Monitoring environment. It provides an overall view of the enterprise network; from this view, the user can "drill down" to examine components of the environment more closely. The Portal includes information from different agents that monitor various parts of the environment; ITCAM Agent for J2EE is one of them.

For details on capabilities of IBM Tivoli Monitoring, and information on deploying the IBM Tivoli Monitoring infrastructure, see *IBM Tivoli Monitoring: Installation and Setup Guide*.

Figure 1 on page 2 shows how ITCAM Agent for J2EE interacts with other IBM Tivoli Monitoring components.

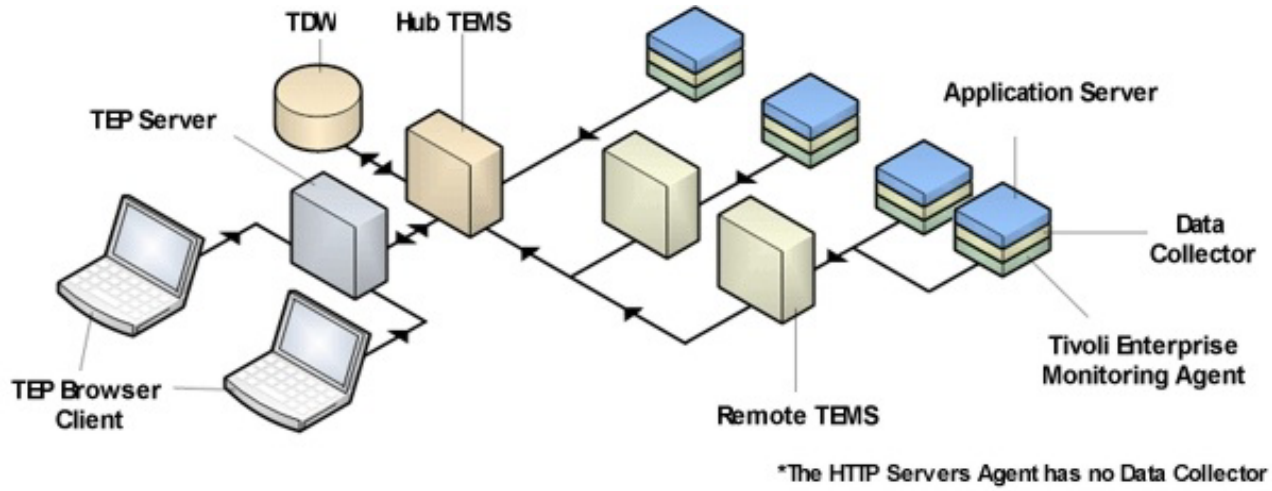

Figure 1. Agent interaction with IBM Tivoli Monitoring

The Managing Server is a component of ITCAM for Application Diagnostics. Its Visualization Engine provides a user interface for "deep dive" diagnostics information. This user interface is a good solution for software developers and performance analysts.

Most information provided by ITCAM Agent for J2EE and available through the Tivoli Enterprise Portal can also be viewed through the Visualization Engine. The Visualization Engine also provides additional diagnostic information, including:

- Method entry/exit and stack tracing,
- Lock analysis,
- Heap object analysis for memory leak diagnosis,
- Thread information,
- "In-flight" request analysis to detect malfunctioning applications.

For details on the capabilities of ITCAM for Application Diagnostics Managing Server, and information on deploying it, see *IBM Tivoli Composite Application Manager for Application Diagnostics Managing Server Installation Guide*.

The diagram on Figure 2 on page 3 shows how ITCAM Agent for J2EE interacts with the components of the Managing Server. (The Data Collector is a component of the Agent).

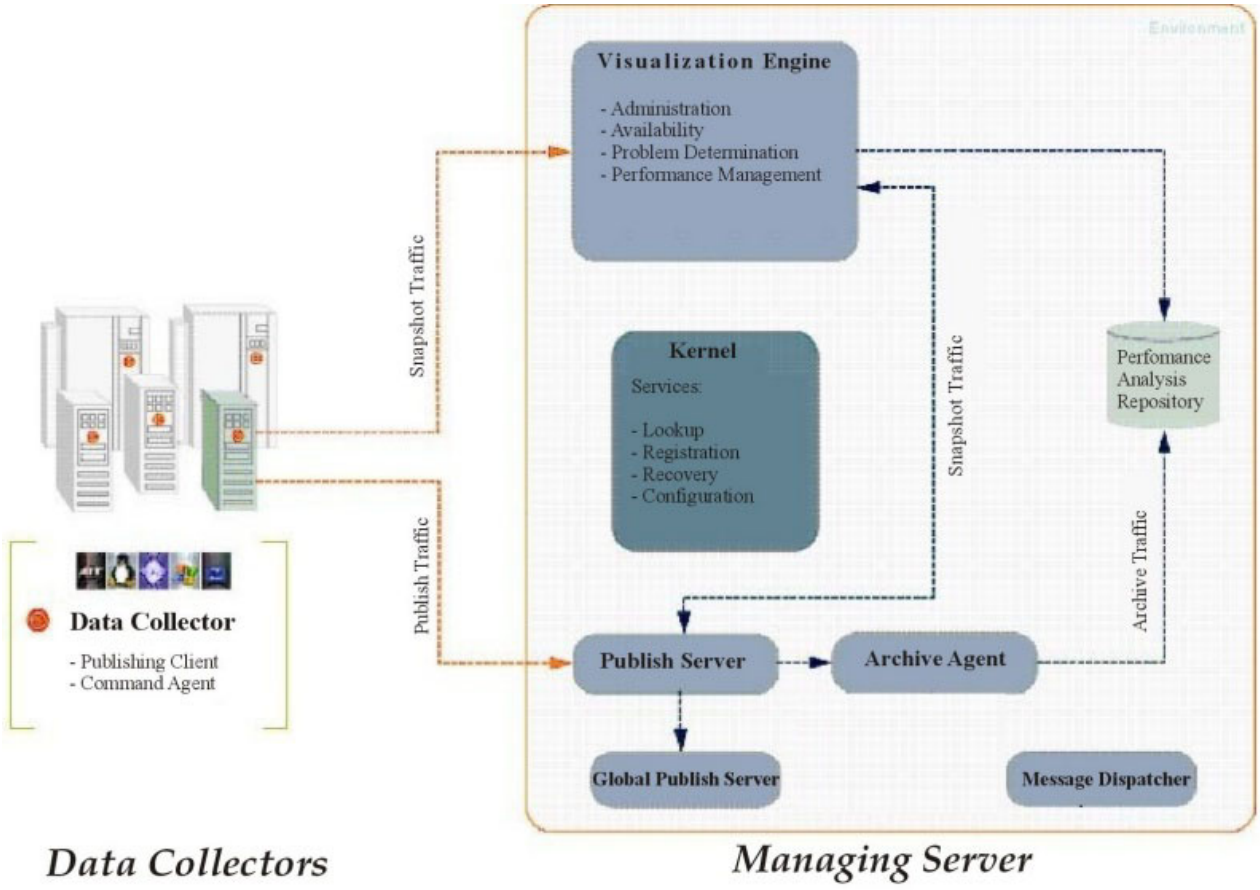

Figure 2. Agent interaction with ITCAM for Application Diagnostics Managing Server

Components of the Agent

ITCAM Agent for J2EE consists of two components: the *Data Collector* and the *Monitoring Agent*. These components are deployed on every monitored host. For interaction with IBM Tivoli Monitoring, the Agent provides *application support files* that are to be installed on servers and clients in the IBM Tivoli Monitoring infrastructure.

Data Collector

The Data Collector collects monitoring and diagnostics information from the application server using the following methods:

Byte Code Instrumentation (BCI)

The data collector injects monitoring calls into the Java code that processes application requests. Data is collected on request processing time and on different types of J2EE API calls within each request.

BCI monitoring requirements can differ. On a production system, request level monitoring might be sufficient; however, on a test or development system, or when a problem is being investigated, BCI can be used to instrument application method entry and exit, synchronized methods, and object allocation. BCI uses a certain amount of system resources, depending

on the amount of injected calls. The level of detail, and thus the use of resources, is determined by the *monitoring level*, which can be set for every monitored application server.

With IBM Tivoli Monitoring, levels L1 and L2 are supported; with ITCAM for Application Diagnostics Managing Server, the additional level L3 is available. The monitoring level can be set for each application server instance, independently on the Tivoli Monitoring components and the managing server.

Java Management Extension (JMX)
Used by most application servers to provide performance monitoring information.

Garbage Collection logs
The logs are written by the application server and contain detailed information about the garbage collection process. Such information can be useful for application monitoring and enhancement. Some application servers provide these logs as files, while others use an API.

The Data Collector sends the information to the monitoring agent. It also communicates directly with the Managing Server (if the Managing Server infrastructure is used).

You must configure the Data Collector for every instance of the application server that you need to monitor.

Monitoring Agent

The monitoring agent collects information from the data collector, processes the information, and aggregates it for presentation to the user. It also parses application server logs. For some application servers, the monitoring agent uses JMX to retrieve performance information; with other application servers, the monitoring agent relies on the data collector for this information.

The Monitoring Agent sends monitoring information to the Tivoli Enterprise Monitoring Server. It also receives Take Action commands from the Tivoli Enterprise Monitoring Server. When these commands involve server management actions (starting, stopping, or restarting the application server), the monitoring agent performs these actions.

Application support files

To enable ITCAM Agent for J2EE interaction with IBM Tivoli Monitoring, the application support files shipped with the Agent **must** be installed on all hub Tivoli Enterprise Monitoring Servers, all Tivoli Enterprise Portal Servers, and Tivoli Enterprise Portal clients except browser-based clients.

On the Tivoli Enterprise Monitoring Server, support files provide the ITCAM Agent for J2EE data tables and situations.

On the Tivoli Enterprise Portal Server, support files provide the ITCAM Agent for J2EE workspaces that display the monitoring information and include code that processes situation information for the Summary workspaces.

On the Tivoli Enterprise Portal client, support files provide the ITCAM Agent for J2EE Helps and Language Packs.

Chapter 2. Prerequisites and pre-installation tasks

Depending on your application server type, you must ensure that certain prerequisites are met and complete certain tasks before installing and configuring ITCAM Agent for J2EE.

Important: The following application servers are not supported by the current version of the data collector:

- Oracle/BEA application server
- WebSphere Application Server Community Edition
- Sun JSAS

Command line length limitations on Windows systems

After installing the Data Collector, several options will be added to the application server startup command. But on Windows systems, there is a limitation on the length of the command line. The overall command length must be less than 8191 characters on Windows XP/2003 and later systems and less than 2047 characters for Windows 2000 systems. Thus, use a path with a short length for the Data Collector installation. For example, for Oracle application server on Windows, use C:\DC as the Data Collector installation path.

Updating configuration of old monitoring agent version

If the Monitoring Agent component of ITCAM for J2EE 6.1 or the J2EE agent of ITCAM for Web Resources 6,.2 is installed on the host, edit the *ITM_home*/config/yj.ini file.

Add the following lines at the end of the file:

```
CTIRA_HOSTNAME=$RUNNINGHOSTNAME$
CTIRA_SYSTEM_NAME=$RUNNINGHOSTNAME$
CTsIRA_NODETYPE='KYJA'
```

Save the file. Then restart the monitoring agent.

Permissions required for installing and configuring ITCAM Agent for J2EE

You can install ITCAM Agent for J2EE as the root or administrator user. Alternatively, if you use a user account without administrator or root privileges. make sure that appropriate permissions are available.

Required permissions common to all J2EE application servers:

- You can use the application server user to install the agent and configure the Data Collector. On UNIX platform (AIX/HP-UX/Linux/Solaris), if the user of the application server and the data collector are not the same, the user of the application server should be a member of the user group for the data collector.
- Read, write, and create new file permissions for the new agent home directory.
- Read permissions to the <AppServer_home> directory and to all subfiles and subdirectories

- Read and Execute File permissions for the JDK directory that is used for starting the application server.

Permissions for the application server after the Data Collector has been installed and successfully configured:

- Read and Write permissions to the garbage collection log file.
- Read, Write, and Create New File permissions to the server-instance-specific runtime directory under <DC_home>/runtime

Note: The permissions for this directory are changed automatically by the Configuration Tool

Table 1. Application-server-specific, required permissions for the user that installs and configures the Data Collector

Application Server	File or Directory	Permissions Required
JBoss	The JBoss startup script file specified by the parameter JBOSSSTARTSH	• Read • Write
JBoss	The JBoss run.jar file at <AppServer_home>/bin/run.jar	• Read
JBoss	The JBoss server instance directory at <AppServer_home>/server/<AppServer_instance>	• Read • Write • Create New File
Tomcat	The Tomcat startup script file specified by the parameter STARTUP_FILE	• Read • Write
Tomcat	The Tomcat configuration file at <AppServer_home>/conf/catalina.properties	• Read • Write
Tomcat	The Tomcat catalina.jar file at <AppServer_home>/lib/catalina.jar	• Read
J2SE	The J2SE application startup script file specified by the parameter J2SESTARTSH	• Read • Write
SAP Netweaver on Linux and UNIX systems	The Netweaver instance startup profile directory. For example: /usr/sap/J2E/SYS/profile/START_JC00_tiv00 In this example, START_JC00_tiv00 is the name of the profile **Note:** The NETWEAVER_INSTANCE_STARTUP_PRFILE file is used for setting the library path and some arguments for the AIX® platform.	• Read • Write
SAP Netweaver	The Central Instance Network Home directory at <central_instance_network_home>/SDM/program	• Read • Write • Create New File
SAP Netweaver	The Central Instance configtool directory at <central_instance_network_home>/j2ee/configtool	• Read • Write • Create New File
WebLogic	The WebLogic startup script file specified by the parameter WL_STARTSH **Note:** This is required only if the WebLogic server instance is started from a script file	• Read • Write

Table 1. Application-server-specific, required permissions for the user that installs and configures the Data Collector (continued)

Application Server	File or Directory	Permissions Required
WebLogic	The WebLogic node manager startup script file at <AppServer_home>/server/bin/startNodeManager.sh(cmd) **Note:** This is required only if the WebLogic server instance is started by the node manager	• Read • Write
WebLogic	WebLogic common environment directory <AppServer_home>/common/bin	• Read • Write
WebLogic	WebLogic startup script file directory. In WebLogic 8, the path is <Domain_Home>, for example, /bea/user_projects/domains/mydomain/ In WebLogic 9, the path is <Domain_Home>/bin, for example, /bea/user_projects/domains/base_domain/bin	• Read • Write
WebLogic on Windows systems	The WebLogic node manager install service file at <AppServer_home>/server/bin/installNodeMgrSvc.cmd **Note:** This is required only if the WebLogic server instance is started by the node manager and the node manager is installed as a Windows service	• Read • Write
WebLogic 9 and 10	The WebLogic common environment file at <AppServer_home>/common/bin/commEnv.sh(cmd) **Note:** This is required only if the WebLogic instance or node manager are started by the WebLogic Script Tool (WST)	• Read • Write

Pre-installation steps for a NetWeaver server

For a NetWeaver server, complete the following pre-installation steps.

1. Manually back up your NetWeaver database. Use the database admin user, such as db2j2e.
2. Make sure the NetWeaver system is running.
3. You need to gather the information of the directories of the **Server Home**, the **Central Instance Home**, and the **Central Instance Network Home**. For detailed information about the directories described, please refer to "Three installation types of ITCAM for J2EE Data Collector for NetWeaver" on page 8.
4. Make sure you have got the system ID and the instance name.
5. For silent installation, use the configtool to get the server ID to be monitored.
6. You should know the Java Naming and Directory Interface (JNDI) port. The JNDI port is a P4 port of the NetWeaver server to be monitored.
7. If distributed dialog instance installation is selected as the installation type, mount the *Central instance home* on central instance computer to a local directory (For example, the absolute path of *Central instance home* on central instance computer on is C:\usr\sap\J2E\JC00, You should map or mount it to a local directory, such as \\<*hostname*>\usr\sap\J2E\JC00 or Y:\usr\sap\J2E\JC00), and make sure you have the writing rights. Where <hostname> is the IP address or qualified host name of the central instance computer.
8. On Linux and UNIX systems the following requirements apply:
   - The admin users for every SAP NetWeaver instance must belong to the same group (for example, sapsys). Run the ITCAM Agent for J2EE installation program as a user belonging to the same group.

- To configure each SAP NetWeaver instance that you need to monitor, run the Data Collector configuration tool using the admin user for the instance.

Important: For paths, always use the backslash (\) on Windows systems and the forward slash (/) on Linux and UNIX systems.

Three installation types of ITCAM for J2EE Data Collector for NetWeaver

The ITCAM for J2EE Data Collector supports three types of installation. Before introducing the three types of installation, be familiar with the following parameters:

- *Server home*: The absolute path of directory wherein the instance is monitored.
- *Central instance home*: The absolute path of central instance home directory.
- *Central instance network home*: A local path mounted from central instance home directory.

1. Central instance installation

Install the ITCAM for J2EE Data Collector to monitor the NetWeaver server on the central instance. Specify the *Server home* for this installation type.

Server home: The absolute path of Central instance home directory (for example, C:\usr\sap\J2E\JC00).

Note: For the silent installation, the value of *Central instance home* and *Central instance network home* should be identical with the value of *Server home*.

2. Local dialog instance installation

Install ITCAM for J2EE DC to monitor the NetWeaver server on the dialog instance which is located on the same server as the central instance is. Specify the *Server home* and *Central instance home* for this installation type.

Server home: The absolute path of local dialog instance home directory (for example, C:\usr\sap\J2E\J01).

Central instance home: The absolute path of central instance home directory (for example, C:\usr\sap\J2E\JC00).

Note: For the silent installation, the value of *Central instance network home* should be identical with *Central instance home*.

3. Distributed dialog instance installation

Install ITCAM for J2EE DC on the dialog instance computer. The central instance is not installed on the same computer as the dialog instance. Specify *Server home*, *Central instance home*, and *Central instance network home* for this installation type.

Server home: The absolute path of distributed dialog instance home directory (for example, C:\usr\sap\J2E\J01).

Central instance home: The absolute path of central instance home directory (for example, C:\usr\sap\J2E\JC00).

Central instance network home: A local path mounted from central instance home directory (for example, Y:\usr\sap\J2E\JC00. This directory is the location where you mounted from the central instance home).

HP-UX and Solaris: Kernel settings for application servers

If you are installing the Data Collector on HP-UX or Solaris (version 9 and older), you need to set the operating system's kernel values to support the application server.

HP-UX

On HP-UX systems, modify several kernel settings.

Several HP-UX kernel values are typically too small for the application server.

Perform the following procedure to adjust the kernel values:

1. Log into the host computer as `root`.
2. Determine the physical memory, which you must know to avoid setting certain kernel parameters above the physical capacity:
   a. Start the HP-UX System Administration Manager (SAM) utility:
      ```
      sam
      ```
      This starts a text-based GUI interface. Use tab and arrow keys to move around in the interface.
   b. Select **Performance Monitors > System Properties > Memory**.
   c. Note the value for Physical Memory and click **OK**.
   d. Exit from the SAM utility.
3. Set the maxfiles and maxfiles_lim parameters to at least `4096`. Table 2 on page 10 shows recommended values of `8000` and `8196`, respectively. You must first edit the /usr/conf/master.d/core-hpux file, so the SAM utility can set values greater than 2048:
   a. Open the /usr/conf/master.d/core-hpux file in a text editor.
   b. Change the line, "*range maxfiles<=2048" to "*range maxfiles<=60000"
   c. Change the line, "*range maxfiles_lim<=2048" to "*range maxfiles_lim<=60000"
   d. Save and close the file. Old values might be stored in the /var/sam/boot.config file. Force the SAM utility to create a new boot.config file:
      1) Move the existing version of the /var/sam/boot.config file to another location, such as the /tmp directory.
      2) Start the SAM utility.
      3) Select Kernel Configuration > Configurable Parameters. When the Kernel Configuration window opens, a new boot.config file exists.

         Alternatively, rebuild the boot.config file with the following command:
         ```
         # /usr/sam/lbin/getkinfo -b
         ```
4. Set new kernel parameter values:
   a. Start the SAM utility.
   b. Click **Kernel Configuration > Configurable Parameters**.
   c. For each of the parameters in the following table, perform this procedure:
      1) Highlight the parameter to change.
      2) Click **Actions > Modify Configurable Parameter**.

3) Type the new value in the Formula/Value field.
4) Click **OK**.

Typical kernel settings for running the application server are displayed in the following table:

Table 2. Typical Kernel settings for Running the Application Server

Parameter	Value
dbc_max_pct	25
maxdsiz	805306358
maxdsiz	2048000000 (when running multiple profiles on the same system)
maxfiles_lim	8196 (Change this one before maxfiles.)
maxfiles	8000
maxssiz	8388608
maxswapchunks	8192
max_thread_proc	3000
maxuprc	512
maxusers	512
msgmap	2048
msgmax	65535
msgmax	131070 (when running multiple profiles on the same system)
msgmnb	65535
msgmnb	131070 (when running multiple profiles on the same system)
msgmni	50
msgseg	32767
msgssz	32
msgtql	2046
nfile	58145
nflocks	3000
ninode	60000
nkthread	7219
nproc	4116
npty	2024
nstrpty	1024
nstrtel	60
sema	1
semaem	16384
semmap	514
semmni	2048
semmns	16384
semmnu	1024
semume	200
semvmx	32767
shmmax	2147483647

Table 2. Typical Kernel settings for Running the Application Server (continued)

Parameter	Value
shmem	1
shmmni	1024
shmseg	1024
STRMSGSZ	65535

Note: When the application server and DB2® are on the same server, some kernel values are higher than those shown in the preceding table.

5. Click **Actions > Process New Kernel**.
6. Click **Yes** on the information window to confirm your decision to restart the server. Follow the on-screen instructions to restart your server and to enable the new settings.
7. If you plan to redirect displays to non-HP servers, complete the following steps before running the application server installation wizard:
   a. Issue the following command to obtain information about all the public locales that are accessible to your application:
      ```
      # locale -a
      ```
   b. Choose a value for your system from the output that is displayed and set the LANG environment variable to this value. Here is an example command that sets the value of LANG to en_US.iso88591:
      ```
      # export LANG=en_US.iso8859
      ```

Solaris

On Solaris systems version 9 and older, modify several kernel settings.

Several Solaris kernel values are typically too small for the application server.

Perform the following procedure to adjust the kernel values:

1. Before installing, review the server configuration:
   ```
   sysdef -i
   ```
   The kernel values are set in the /etc/system file, as shown in the following example.
   ```
   set shmsys:shminfo_shmmax = 4294967295
   set shmsys:shminfo_shmseg = 1024
   set shmsys:shminfo_shmmni = 1024
   set semsys:seminfo_semaem = 16384
   set semsys:seminfo_semmni = 1024
   set semsys:seminfo_semmap = 1026
   set semsys:seminfo_semmns = 16384
   set semsys:seminfo_semmsl = 100
   set semsys:seminfo_semopm = 100
   set semsys:seminfo_semmnu = 2048
   set semsys:seminfo_semume = 256
   set msgsys:msginfo_msgmap = 1026
   set msgsys:msginfo_msgmax = 65535
   set rlim_fd_cur=1024
   ```
2. Change kernel values by editing the /etc/system file then rebooting the operating system.
   For more information about setting up the Solaris system, see the Solaris System Administration documentation at the following Web site: http://docs.sun.com/app/docs/prod/solaris.admin.misc

For example, the *Solaris Tunable Parameters Reference Manual* at the following Web site: http://docs.sun.com/app/docs/doc/816-7137?q=shmsys

Queue managers are generally independent of each other. Therefore system kernel parameters, for example shmmni, semmni, semmns, and semmnu need to allow for the number of queue managers in the system.

Chapter 3. Installing and uninstalling ITCAM Agent for J2EE on Windows

Use the installer utility to install ITCAM Agent for J2EE on every monitored Windows host. The installer instals both the monitoring agent and the data collector.

You can also use the silent mode of the installer utility. Silent mode can be convenient for speedy installation on many hosts.

You must also install application support files for the agent on the Tivoli Enterprise Portal Server and every hub Tivoli Enterprise Monitoring server.

Tip: You can also use Tivoli Monitoring to install the agent remotely. For instructions, see the "Deploying monitoring agents in your environment" topic in the IBM Tivoli Monitoring Installation and Setup Guide.

Installing the agent using the Windows installation utility

To install the agent, run the Windows installation utility.

Before you begin

Before installing the Agent, you need to know the host name or IP address of the Tivoli Enterprise Monitoring Server to which the agent is to connect.

Procedure

1. Extract the agent installation image.
2. Run the `setup.exe` file.
3. Use the installation wizard to install the agent:
   a. Review prerequisites.
   b. Review and accept the product license.
   c. Select the folder for installation.
   d. If Tivoli Monitoring is not installed on the host, enter the 32-character encryption key used to secure password transmission and other sensitive data across your IBM Tivoli Monitoring environment. For more details about the key, see the *IBM Tivoli Monitoring: Installation and Setup Guide*.
   e. Select the components to install.

      Important: If any IBM Tivoli Monitoring Agent is already installed on this host, make sure to expand the tree in this window and explicitly select the **IBM Tivoli Composite Application Manager Agent for J2EE** check box. By default, if an IBM Tivoli Monitoring Agent is found, this check box is not selected even when you select the top level box.
   f. Enter the name of the Windows program folder. After completion of the installation, a folder with this name appears in your **Start** > **Programs** menu. The folder contains Tivoli Monitoring utilities.
   g. Verify the selected features and install the agent.
   h. Select whether to configure the monitoring agent settings and whether to launch the Manage Tivoli Monitoring Services utility for additional

configuration. For configuration instructions, see Chapter 5, "Configuring and unconfiguring the monitoring agent and data collector," on page 33.

Installing application support on Windows

To ensure ITCAM Agent for J2EE works within your IBM Tivoli Monitoring infrastructure, you need to install application support files for it on every hub monitoring server, portal server, and portal client. After configuring the Agent on the monitored host, you also need to enable Tivoli monitoring history collection. You do not need to install application support files if IBM Tivoli Monitoring is not used (in a deep dive diagnostics only installation).

Important: You will need to stop the monitoring server, portal server, or portal client when installing the support files.

Installing application support on the Tivoli Enterprise Monitoring Server

1. Stop the Tivoli Enterprise Monitoring Server. The installer automatically stops the Tivoli Enterprise Monitoring Server; you can also choose to stop the server manually before starting the installer. Perform the following steps to stop the Tivoli Enterprise Monitoring Server manually:
   a. Click **Start** > **Programs** > **IBM Tivoli Monitoring** > **Manage Tivoli Monitoring Services**.
   b. Right-click Tivoli Enterprise Monitoring Server.
   c. In the pop-up menu, select **Stop**.
2. Access the `\WINDOWS` subdirectory on the agent installation image.
3. Double-click **`setup.exe`**.
4. Click **Next** on the Welcome window.
5. The Software License Agreement window is displayed. Select **I accept the terms in the license agreement** and click **Next**.
6. Select **Tivoli Enterprise Monitoring Server - TEMS** and click **Next**.

   Note: If you have other components installed on the same computer, such as the desktop client, also select those components to install the component-specific application support.
7. Review the installation summary details. Click **Next** to start the installation.
8. Select the setup type that best suits your needs.

   In the following steps you will be promptd for the information required to configure the items that are listed in the **Setup Type** window. You can uncheck the box to delay the configuration until the installation is complete. Some configuration items are mandatory (preceded by an *) and cannot be unchecked.
9. Specify the location of the monitoring server. Choose **On this computer** to install application support on the host you are running the setup file on, and **On a different computer** otherwise. Then click **OK**.
10. Select the application support to add to the monitoring server and click **OK**. By default, application supports which are not yet installed on this server are selected.
11. Review the application support addition details and click **Next**.
12. Specify the default values for the agent to use when it communicates with the monitoring server and click **OK**.

Note:

- You can specify three methods for communication to set up backup communication methods. If the method you have identified as Protocol 1 fails, Protocol 2 is used.
- You can specify the default values for a backup communication between the agent and the monitoring server by selecting **Option Secondary TEMS Connection**.

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

b. Identify the type of protocol that the agent uses to communicate with the monitoring server. You have five choices: `IP.UDP`, `IP.PIPE`, `IP.SPIPE`, `SNA`, `No TEMS`.

13. Define the communications between agents and the monitoring server and click **OK**.
14. Click **Finish**.

Installing application support on the Tivoli Enterprise Portal Server

1. Open **Manage Tivoli Enterprise Monitoring Services.**
2. Stop the portal server by right-clicking it and clicking **Stop**.
3. Access the `\WINDOWS` subdirectory on the agent installation media.
4. Double-click **`setup.exe`**.
5. Click **Next** on the Welcome window.
6. The Software License Agreement window is displayed. Select **I accept the terms in the license agreement** and click **Next**.
7. Select **Tivoli Enterprise Portal Server - TEPS** and click **Next**.

   Note: If you have other components installed on the same computer, such as the desktop client, also select those components to install the component-specific application support.
8. If you need remote configuration in the future, select the agent to add it to the remote deployment depot, and click **Next**. Otherwise, click **Next** without selecting any agents.
9. Review the installation summary details. Click **Next** to start the installation.
10. Select the setup type that best suits your needs.

    In the following steps you will be promptd for the information required to configure the items that list in the **Setup Type** window. You can uncheck the box to delay the configuration until the installation is complete. Some configuration items are mandatory (preceded by an *) and cannot be unchecked.
11. Type the host name for the portal server and click **Next**.
12. Click **Finish**.
13. Restart the portal server.

Important: If the Tivoli Enterprise Portal Server provides the browser client, check that the Eclipse help server has been configured. See “Ensuring that the Eclipse server has been configured” on page 16.

Installing application support on the Tivoli Enterprise Portal desktop client

1. Stop the desktop client before performing this procedure.
2. Access the \WINDOWS subdirectory on the agent installation media.
3. Double-click **setup.exe**.
4. Click **Next** on the Welcome window.
5. The Software License Agreement window is displayed. Select **I accept the terms in the license agreement** and click **Next**.
6. Select **TEP Desktop Client - TEPD** and click **Next**.
7. If you need remote configuration in the future, select the agent to add it to the remote deployment depot, and click **Next**. Otherwise, click **Next** without selecting any agents.
8. Review the installation summary details. Click **Next** to start the installation.
9. Select the setup type that best suits your needs.

   In the following steps you will be promptd for the information required to configure the items that list in the **Setup Type** window. You can uncheck the box to delay the configuration until the installation is complete. Some configuration items are mandatory (preceded by an *) and cannot be unchecked.
10. Type the host name for the portal server and click **Next**.
11. Click **Finish** to complete the installation.

Important: Check that the Eclipse help server has been configured for the client. See “Ensuring that the Eclipse server has been configured.”

Ensuring that the Eclipse server has been configured

After installing application support files on a Tivoli Enterprise Portal Server that provides the browser client or on a Tivoli Enterprise Portal desktop client, you must check the Eclipse help server for the portal client to ensure that it has been configured.

Start Manage Tivoli Enterprise Monitoring Services (**Start** > **All Programs** > **IBM Tivoli Monitoring** > **Manage Tivoli Monitoring Services**), and ensure that the **Eclipse Help Server** entry indicates **Yes** in the Configured column.

If the entry indicates **No**, you must configure the Eclipse server. To do this, right-click the entry, and select **Configure Using Defaults** from the pop-up menu.

You are prompted for the port number that the Eclipse Help Server will use. Ensure that this value is set to the same port number that you specified when installing IBM Tivoli Monitoring, and click **OK**.

If you want the Eclipse help server to start automatically whenever this node is started, right-click the **Eclipse Help Server** entry, and select **Change Startup** from the pop-up menu. The Eclipse server's startup parameters are displayed. Select **Automatic** in the **startup type** field, and click **OK**.

Performing a silent installation or uninstallation on Windows

You can use the Installer to install or uninstall ITCAM Agent for J2EE in silent mode. You can also install or uninstall support files for the Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, and Tivoli Enterprise Portal client on Windows in silent mode. To do this, modify the sample files provided on the installation image, and then run the installer from the command line.

To perform a silent installation or uninstallation, first you need to prepare the response file. Then, run the installer, supplying the name of the response file.

Preparing the response file for Agent installation

To prepare a response file for installing the Agent, perform the following procedure:

1. On the product installation image, in the `WINDOWS\Deploy` directory, locate the `YJ_Silent_Install.txt` file.
2. Make a copy of this file, and open it in a text editor.
3. Modify any of the following properties, if necessary. Do not modify any other properties.

Table 3. Agent installation response file properties

Response file property	Meaning
Install Directory	The directory (*ITM_home*) where the Agent is to be installed. The destination directory can be shared with other IBM Tivoli Monitoring products. If you want to use a location other than the default (`C:\IBM\ITM`), click **Browse**, and select the folder that you want to use. **Note:** You can have multiple installations of the Agent on the same host. In this case, specify a different destination folder for each installation.
Install Folder	The Windows program folder (under the **Programs** menu) where IBM Tivoli Monitoring programs will be listed.
EncryptionKey	The 32-character encryption key used to secure password transmission and other sensitive data across your IBM Tivoli Monitoring environment. See IBM Tivoli Monitoring: Installation and Setup Guide for details about the encryption key.

4. Save the edited copy in a work directory, for example, as `C:\TEMP\SILENT.TXT`.

Preparing the response file for Agent uninstallation

To prepare a response file for uninstalling the Agent, perform the following procedure:

1. On the product installation image, in the `WINDOWS\Deploy` directory, locate the `YJ_Silent_Uninstall.txt` file.
2. Copy the file to a work directory, for example, as `C:\TEMP\SILENT.TXT`. Do not modify the copy.

Preparing the response file for Support Files installation

To prepare a response file for installing the support files on a Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, or Tivoli Enterprise Portal client, perform the following procedure:

1. On the product installation image, in the WINDOWS directory, locate the silent.txt file.
2. Make a copy of this file, and open it in a text editor.
3. Find the following lines and uncomment (by removing ; as the first character) those that apply to the host you are installing on:

   ```
   KYJWICMS=ITCAM Agent for WebSphere Applications Support ( TEMS )
   KYJWIXEW=ITCAM Agent for WebSphere Applications Support ( TEP Workstation )
   KYJWICNS=ITCAM Agent for WebSphere Applications Support ( TEP Server )
   ```

4. Save the edited copy in a work directory, for example, as C:\TEMP\SILENT.TXT.

Preparing the response file for Support Files uninstallation

To prepare a response file for uninstalling the support files on a Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, or Tivoli Enterprise Portal client, perform the following procedure:

1. On the product installation image, in the WINDOWS directory, locate the silent.txt file.
2. Make a copy of this file, and open it in a text editor.
3. Find the following line and uncomment it (by removing ; as the first character):

   ```
   UNINSTALLSELECTED=Yes
   ```

4. Find the following lines and uncomment (by removing ; as the first character) those that apply to the host you are uninstalling on:

   ```
   KYJWICMS=ITCAM Agent for WebSphere Applications Support ( TEMS )
   KYJWIXEW=ITCAM Agent for WebSphere Applications Support ( TEP Workstation )
   KYJWICNS=ITCAM Agent for WebSphere Applications Support ( TEP Server )
   ```

5. Save the edited copy in a work directory, for example, as C:\TEMP\SILENT.TXT.

Running the Installer in silent mode

After preparing the response file for your installation and uninstallation, run the installer, specifying the path and name for the response file. Perform the following procedure:

1. Open a Windows command prompt window, and change to the WINDOWS directory on the installation image.
2. Invoke setup as follows. Specify the parameters in the exact order shown:

   ```
   start /wait setup /z"/sfresponse_file_name" /s /f2"log_file_name"
   ```

   where *response_file_name* is the name of the response file you have prepared (with full path), and *log_file_name* is the name of the log file that the Installer will write (with full path). For example:

   ```
   start /wait setup /z"/sfC:\TEMP\SILENT.TXT" /s /f2"C:\TEMP\INSTALL.LOG"
   ```

Attention: if you are performing an upgrade or maintenance level update, and the Monitoring Agent is currently running, silent installation will be aborted.

You can find complete information about silent IBM Tivoli Monitoring installation in "Appendix B. Performing a silent installation of IBM Tivoli Monitoring" of the *IBM Tivoli Monitoring: Installation and Setup Guide*.

Installing and uninstalling a Language Pack on Windows

A Language Pack enables user interaction with the agent in a language other than English. For example, when a Spanish language pack is installed, the Tivoli Enterprise Portal workspaces and the internal messages of the Agent are displayed in Spanish.

To enable full support for a language, you must install the Language Pack on the agent host and all hosts where the Tivoli monitoring support files for the agent are installed (hub Tivoli Enterprise Monitoring Servers, all Tivoli Enterprise Portal Servers, and all Tivoli Enterprise Portal desktop clients).

If you no longer want to use a language, uninstall the language pack for it.

Before installing or uninstalling a Language Pack, ensure that:

- The agent and the Tivoli Enterprise Portal Support Files are installed.
- The Java runtime environment (JRE) is available on every host where you are planning to install the Language Pack. (The JRE is required by IBM Tivoli Monitoring).

Installing a Language Pack on Windows

To install a Language Pack on Windows you need to use the installer on the Language Pack image. The procedure is the same on the Agent host, hub Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, and Tivoli Enterprise Portal desktop client.

Perform the following procedure:

1. Start `lpinstaller.exe` from the Language Pack image.
2. Select the language of the installer and click OK.

   Note: In this step, you select the language for the installer user interface, not the language pack that will be installed.
3. Click **Next** on the Introduction window.
4. Select **Add/Update** and click **Next**.
5. Select the folder where the National Language Support package (NLSPackage) files are located. This is the `nlspackage` folder on the Language Pack DVD.
6. Select **ITCAM Agent for J2EE**.
7. Select the languages to install and click **Next**.

   Note: You can hold down the **Ctrl** key for multiple selections.
8. Examine the installation summary page and click **Next** to begin installation.
9. Click **Next**.
10. Click **Finish** to exit the installer.
11. If you are installing the Language Pack on a Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, or Tivoli Enterprise Portal desktop client, start the **Manage Tivoli Monitoring Services** utility, and use it to restart the server or client. If the Eclipse Help Server is running, restart it as well.

Uninstalling a Language Pack on Windows

To uninstall a Language Pack on Windows you need to use the installer on the Language Pack image. The procedure is the same on the Agent host, hub Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, and Tivoli Enterprise Portal desktop client.

Perform the following procedure:

1. Start `lpinstaller.exe` from the Language Pack image.
2. Select the language of the installer and click OK.

   Note: In this step, you select the language for the installer user interface, not the language pack that will be installed.
3. Click **Next** on the Introduction window.
4. Select **Remove** and click **Next**.
5. Select **ITCAM Agent for J2EE**.
6. Select the languages to uninstall and click **Next**.

   Note: You can hold down the **Ctrl** key for multiple selections.
7. Examine the uninstallation summary page and click **Next** to begin installation.
8. Click **Next**.
9. Click **Finish** to exit the installer.
10. If you are uninstalling the Language Pack on a Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, or Tivoli Enterprise Portal desktop client, start the **Manage Tivoli Monitoring Services** utility, and use it to restart the server or client. If the Eclipse Help Server is running, restart it as well.

Uninstalling ITCAM Agent for J2EE on Windows

To remove ITCAM Agent for J2EE on Windows systems, first unconfigure the Data Collector from all application server instances. For instructions, see "Unconfigure the Data Collector from application server instances" on page 41.

When the Data Collector is unconfigured, perform the following procedure:

1. From the desktop, click **Start → Settings → Control Panel** (for Windows 2000) or **Start → Control Panel** (for Windows 2003).
2. Click **Add or Remove Programs**.
3. Select **IBM Tivoli Monitoring**.
4. Click **Change**.
5. Perform one of the following procedures:
   - If you want to remove all IBM Tivoli Monitoring components, including the Agent, select **Remove** and click **Next**. Click **OK** to confirm the uninstallation.
   - If you want to remove the Agent but not other IBM Tivoli Monitoring components, select **Modify** and click **Next**. Deselect the Agent and click**Next** several times to complete the uninstallation.
6. Click **Finish**.

Chapter 4. Installing ITCAM Agent for J2EE on Linux and UNIX systems

Use the installer utility to instal ITCAM Agent for J2EE on every monitored Linux and UNIX host. The installer instals both the monitoring agent and the data collector.

You can also use the silent mode of the installer utility. Silent mode can be convenient for speedy installation on many hosts.

You must also install application support files for the agent on the Tivoli Enterprise Portal Server and every hub Tivoli Enterprise Monitoring server.

Tip: You can also use Tivoli Monitoring to install the agent remotely. For instructions, see the "Deploying monitoring agents in your environment" topic in the IBM Tivoli Monitoring Installation and Setup Guide.

Installing the agent using the command line installation utility

To install the agent, run the command line installation utility.

Before you begin

Before installing the Agent, you need to know the host name or IP address of the Tivoli Enterprise Monitoring Server to which the agent is to connect.

Procedure

1. Extract the agent installation image.
2. Run the `./install.sh` script.
3. Use the installation wizard to install the agent:
   a. Enter the directory for installation.
   b. When prompted for installation options, select 1, `Install products to the local host`. ITCAM Agent for J2EE does not support other options.
   c. Review and accept the product license.
   d. If Tivoli Monitoring is not installed on the host, enter the 32-character encryption key used to secure password transmission and other sensitive data across your IBM Tivoli Monitoring environment. For more details about the key, see the *IBM Tivoli Monitoring: Installation and Setup Guide*.
   e. If the installation utility requests installation of additional prerequisites, for example `IBM Tivoli Monitoring Shared Libraries`, select their installation. If you choose not to install the prerequisites, the utility does not install the agent.
   f. When prompted to select the product to install, select `IBM Tivoli Composite Application Manager Agent for J2EE`.
   g. Verify the selected features and install the agent.
   h. After all of the components are installed, you are asked whether you want to install additional products or product support packages. Type 2 and press Enter.

i. If your Tivoli Monitoring environment is not already secured you will be asked at this point if you want to secure it. If your Tivoli Monitoring environment is already secured this question is skipped. The product installation process creates the majority of directories and files with world write permissions. IBM Tivoli Monitoring provides the secureMain utility to help you keep the monitoring environment secure. You can secure your installation now, or manually execute the secureMain utility later. For more information, see "Securing your IBM Tivoli Monitoring installation on Linux or UNIX" in the *IBM Tivoli Monitoring Installation Guide, Version 6.2.3*.

Deep dive diagnostics only installation: disabling Monitoring Agent autostart

If you are performing a deep dive diagnostics only installation, where IBM Tivoli Monitoring is not used, disable Monitoring Agent autostart. Do not disable it if Tivoli Monitoring is used.

To disable Monitoring Agent autostart, perform the following procedure:

1. Check the contents of the file *ITM_home*/registry/AutoStart, and get the number from that file. Use this number as *NUM* in the following step.
2. Edit the autostart file for the operating system:
   - On AIX: /etc/rc.itm*NUM*
   - On HP-UX: /sbin/init.d/ITMAgents*NUM*
   - On Linux: /etc/init.d/ITMAgents*NUM*
   - On Solaris: /etc/init.d/ITMAgents*NUM*

   In this file, find and comment out (using the # symbol) the lines with the itmcmd agent start yj and itmcmd agent stop yj commands.

   Example:

```
start_all()
{
/bin/su - root -c " /opt/IBM/YJ1024/bin/itmcmd agent start yj >/dev/null 2>&1"
}

stop_all()
{
/bin/su - root -c " /opt/IBM/YJ1024/bin/itmcmd agent stop yj >/dev/null 2>&1"
}
```

   In this example, you need to comment out both lines starting with /bin/su.

Additional procedure for Security Enhanced Linux (SELinux)

After installing ITCAM Agent for J2EE on SELinux, for example Red Hat Enterprise Linux Version 5 or SUSE Linux Enterprise Server Version 11, you must perform an additional procedure to identify the Data Collector shared libraries.

To identify the Data Collector shared libraries on SELinux, run the following command as root, substituting the installation directory for *ITM_HOME* and the Tivoli Monitoring architecture identifier for *TEMA_architecture_code* and *DC_architecture_code*:

```
chcon -R -t texrel_shlib_t
  ITM_HOME/TEMA_architecture_code/yn/wasdc/7.1.0.2/toolkit/lib/DC_architecture_code
```

The architecture code identifiers for Linux systems are:

- li6263: Linux Intel R2.6 (32 bit)
- lx8266: Linux x86_64 R2.6 (64 bit)
- lpp263: Linux ppc R2.6 (32 bit)
- lpp266: Linux ppc R2.6 (64 bit)
- ls3263: Linux S390 R2.6 (32 bit)
- ls3266: Linux S390 R2.6 (64 bit)

For 32-bit systems, use the architecture code for the system for both *TEMA_architecture_code* and *DC_architecture_code*.

For 64-bit systems, you need to run the command two times in order to identify shared libraries for both 32-bit and 64-bit versions of the Data Collector.

For 64-bit ppc systems, the Monitoring Agent is always 32-bit. Therefore, use lpp263 as *TEMA_architecture_code*. For example:

```
chcon -R -t texrel_shlib_t /opt/ibm/itm/lpp263/yn/wasdc/7.1.0.2/toolkit/lib/lpp263
chcon -R -t texrel_shlib_t /opt/ibm/itm/lpp263/yn/wasdc/7.1.0.2/toolkit/lib/lpp266
```

For 64-bit S390 systems, the Monitoring Agent is always 32-bit. Therefore, use ls3263 as *TEMA_architecture_code*. For example:

```
chcon -R -t texrel_shlib_t /opt/ibm/itm/ls3263/yn/wasdc/7.1.0.2/toolkit/lib/ls3263
chcon -R -t texrel_shlib_t /opt/ibm/itm/ls3263/yn/wasdc/7.1.0.2/toolkit/lib/ls3266
```

For x86_64 systems, the Monitoring Agent might be either 32-bit or 64-bit. Therefore, you need to use li6263 or lx8266 as *TEMA_architecture_code*, depending on which directory exists: *ITM_HOME*/li6263 or *ITM_HOME*/lx8266. You can safely try both sets of commands; the commands applying to a non-existent directory will fail. For example:

```
chcon -R -t texrel_shlib_t /opt/ibm/itm/li6263/yn/wasdc/7.1.0.2/toolkit/lib/li6263
chcon -R -t texrel_shlib_t /opt/ibm/itm/li6263/yn/wasdc/7.1.0.2/toolkit/lib/lx8266

chcon -R -t texrel_shlib_t /opt/ibm/itm/lx8266/yn/wasdc/7.1.0.2/toolkit/lib/li6263
chcon -R -t texrel_shlib_t /opt/ibm/itm/lx8266/yn/wasdc/7.1.0.2/toolkit/lib/lx8266
```

Installing application support on Linux and UNIX systems

To ensure that ITCAM Agent for J2EE works within your IBM Tivoli Monitoring infrastructure, you need to install application support files for it on every hub monitoring server, portal server, and portal client. After configuring the Agent on the monitored host, you also need to enable Tivoli monitoring history collection. You do not need to install application support files if IBM Tivoli Monitoring is not used (in a deep dive diagnostics only installation).

Important: You will need to stop the monitoring server, portal server, or portal client when installing the support files.

Installing application support on the Tivoli Enterprise Monitoring Server

1. Stop the monitoring server by running the following command:

   ```
   ./itmcmd server stop tems_name
   ```

2. Run **./install.sh** from the installation media
3. Press **Enter** to accept the default directory (/opt/IBM/ITM) or type the full path to the installation directory you used when the software asks for the IBM Tivoli Monitoring home directory.

The software displays the following prompt:

```
Select one of the following:
1) Install products to the local host.
2) Install products to depot for remote deployment (requires TEMS).
3) Install TEMS support for remote seeding
4) Exit install.

Please enter a valid number:
```

4. Type **1** and press **Enter**.
5. The software license agreement is displayed after the initialization, enter 1 to accept the agreement and press **Enter**.
6. Type the 32 character encryption key that was specified during the installation of the monitoring server and press **Enter**.

   Note: If you have already installed another IBM Tivoli Monitoring component on this computer or you are installing support for an agent from an agent installation image, this step does not occur.

   The information of installed products is displayed.
7. Press **Enter** to continue the installation and the installer propmts you with the following message:

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

    1) Tivoli Enterprise Portal Browser Client support
    2) Tivoli Enterprise Portal Desktop Client support
    3) Tivoli Enterprise Portal Server support
    4) Tivoli Enterprise Monitoring Server support

   Type the number for the OS you want, or type "q" to quit selection:
   ```
8. Type **4** and press **Enter** to install the application support on the Tivoli Enterprise Monitoring server and the following message is displayed:

   ```
   You selected number "4" or "Tivoli Enterprise Monitoring Server support"

   Is the selection correct [ 1=Yes, 2=No; default is "1"]?
   ```
9. Type **1** and press **Enter** to confirm the selection. The message about the products to install is displayed, for example:

   ```
   The following products are available for installation:

    1) IBM Tivoli Composite Application Manager Agent for J2EE
      v07.10.01.00
    2) all of the above

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

   Type you selections here:
   ```
10. Type the number corresponding to `all of the above` and press **Enter** and the installer prompts you with the following message to ask you to confirm your selection:

    ```
    The following products will be installed:

     IBM Tivoli Composite Application Manager Agent for J2EE
       V07.10.01.00

    Are your selections correct [ 1=Yes, 2=No; default is "1" ]?
    ```
11. Type **1** and press **Enter** to confirm your selection and start the installation.

12. After installing all of the components, the following message is displayed to ask you whether you want to install components for a different operating system:

    ```
    Do you want to install additional products or product support packages
     [ 1=Yes, 2=No; default is "2" ]?
    ```

    Type **2** and press **Enter**.
13. The installation step completes and the information of installed Tivoli Enterprise Monitoring Server product supports is displayed:

    ```
    *) IBM Tivoli Composite Application Manager Agent for J2EE
    ```

    And the installer also prompts you with the following message to seed product supports on the Tivoli Enterprise Monitoring Server:

    ```
    Note: This operation causes the monitoring server to restart.
    Do you want to seed product support on the Tivoli Enterprise Monitoring Server?
    [ 1=Yes, 2=No; default is "1" ]?
    ```
14. Press **Enter** to make the default choice.
15. After starting the Tivoli Enterprise Monitoring Server, the message about the application supports to seed is displayed:

    ```
    The following new Tivoli Enterprise Monitoring Server product support packages
    will be seeded:
     *) IBM Tivoli Composite Application Manager Agent for J2EE

    Select listed above Tivoli Enterprise Monitoring Server product support
    for which default distribution list will be upgraded:
    [1=new, 2=all, 3=none] (Default is: 1):
    ```
16. Press **Enter** to make the default choice.
17. After the support seeding and stopping the monitoring server, the following message is displayed to remind you about the configuration:

    ```
    You may now configure any locally intalled IBM Tivoli Monitoring product via
    the "/opt/IBM/ITM/bin/itmcmd config" command.
    ```
18. The monitoring server is restarted automatically.

Installing application support on the Tivoli Enterprise Portal Server

On a Tivoli Enterprise Portal Server, you must install application support files both for the server itself and for the browser client.

Stop the portal server before performing this procedure.

1. Run **./install.sh** from the installation media
2. Press **Enter** to accept the default directory (/opt/IBM/ITM) or type the full path to the installation directory you used when the software asks for the IBM Tivoli Monitoring home directory.

   The software displays the following prompt:

   ```
   Select one of the following:
   1) Install products to the local host.
   2) Install products to depot for remote deployment (requires TEMS).
   3) Install TEMS support for remote seeding
   4) Exit install.

   Please enter a valid number:
   ```
3. Type **1** and press **Enter**.
4. The software license agreement is displayed after the initialization, enter 1 to accept the agreement and press **Enter**.

5. Type the 32 character encryption key that was specified during the installation of the monitoring server and press **Enter**.

   Note: If you have already installed another IBM Tivoli Monitoring component on this computer or you are installing support for an agent from an agent installation image, this step does not occur.

   The information of installed products is displayed.
6. Press **Enter** to continue the installation and the installer prompts you with the following message:

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

 1) Tivoli Enterprise Portal Browser Client support
 2) Tivoli Enterprise Portal Desktop Client support
 3) Tivoli Enterprise Portal Server support
 4) Tivoli Enterprise Monitoring Server support

Type the number for the OS you want, or type "q" to quit selection:
```

7. Type **3** and press **Enter** to install the application support on the Tivoli Enterprise Portal server and the following message is displayed:

```
You selected number "3" or "Tivoli Enterprise Portal Server support"

Is the selection correct [ 1=Yes, 2=No; default is "1"]?
```

8. Type **1** and press **Enter** to confirm the selection and the message about the products to install is displayed:

```
The following products are available for installation:

 1) IBM Tivoli Composite Application Manager Agent for J2EE
     v07.10.01.00
 2) all of the above

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

Type you selections here:
```

9. Type the number corresponding to `all of the above` and press **Enter** and the installer prompts you with the following message to ask you to confirm your selection:

```
The following products will be installed:

 IBM Tivoli Composite Application Manager Agent for J2EE
    V07.10.01.00

Are your selections correct [ 1=Yes, 2=No; default is "1" ]?
```

10. Type **1** and press **Enter** to confirm your selection and start the installation.
11. After installing all of the components, the following message is displayed to ask you whether you want to install components for a different operating system:

```
Do you want to install additional products or product support
   packages [ 1=Yes, 2=No; default is "2" ]?
```

    Type **1** and press **Enter**.
12. The following message is displayed.

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

 1) Tivoli Enterprise Portal Browser Client support
```

```
 2) Tivoli Enterprise Portal Desktop Client support
 3) Tivoli Enterprise Portal Server support
 4) Tivoli Enterprise Monitoring Server support

Type the number for the OS you want, or type "q" to quit selection:
```

13. Type **1** and press **Enter** to install the application support on the Tivoli Enterprise Portal browse client and the following message is displayed:

```
You selected number "1" or "Tivoli Enterprise Portal Browse Client support"

Is the selection correct [ 1=Yes, 2=No; default is "1"]?
```

14. Type **1** and press **Enter** to confirm the selection and the message about the products to install is displayed:

```
The following products are available for installation:

 1) IBM Tivoli Composite Application Manager Agent for J2EE
   v07.10.00.01
 2) all of the above

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

Type you selections here:
```

15. Type the number corresponding to `all of the above` and press **Enter** and the installer prompts you with the following message to ask you to confirm your selection:

```
The following products will be installed:

 IBM Tivoli Composite Application Manager Agent for J2EE
   V07.10.01.00

Are your selections correct [ 1=Yes, 2=No; default is "1" ]?
```

16. Type **1** and press **Enter** to confirm your selection and start the installation.
17. After installing all of the components, the following message is displayed to ask you whether you want to install other components:

```
Do you want to install additional products or product support packages
 [ 1=Yes, 2=No; default is "2" ]?
```

    Type **2** and press **Enter**.

18. The installation program will complete the installation and exit. After this, re-configure the portal server and browser client by running:

```
itmcmd config -A cq
```

    At any prompts, press **Enter** to accept the default values.

Important: If the Tivoli Enterprise Portal Server provides the browser client, check that the Eclipse help server has been configured. See "Ensure that the Eclipse server has been configured" on page 29.

Installing application support on the Tivoli Enterprise Portal desktop client

Note: Stop the desktop client before performing this procedure.

1. Run **./install.sh** from the installation media
2. Press **Enter** to accept the default directory (`/opt/IBM/ITM`) or type the full path to the installation directory you used when the software asks for the IBM Tivoli Monitoring home directory.

The software displays the following prompt:

```
Select one of the following:
1) Install products to the local host.
2) Install products to depot for remote deployment (requires TEMS).
3) Install TEMS support for remote seeding
4) Exit install.

Please enter a valid number:
```

3. Type **1** and press **Enter**.
4. The software license agreement is displayed after the initialization, enter 1 to accept the agreement and press **Enter**.
5. Type the 32 character encryption key that was specified during the installation of the monitoring server and press **Enter**.

   Note: If you have already installed another IBM Tivoli Monitoring component on this computer or you are installing support for an agent from an agent installation image, this step does not occur.

   The information of installed products is displayed.
6. Press **Enter** to continue the installation and the installer prompts you with the following message:

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

 1) Tivoli Enterprise Portal Browser Client support
 2) Tivoli Enterprise Portal Desktop Client support
 3) Tivoli Enterprise Portal Server support
 4) Tivoli Enterprise Monitoring Server support

Type the number for the OS you want, or type "q" to quit selection:
```

7. Type **2** and press **Enter** to install the application support on the Tivoli Enterprise Portal desktop client and the following message is displayed:

```
You selected number "2" or "Tivoli Enterprise Portal Desktop Client support"

Is the selection correct [ 1=Yes, 2=No; default is "1"]?
```

8. Type **1** and press **Enter** to confirm the selection and the message about the products to install is displayed:

```
The following products are available for installation:

 1) IBM Tivoli Composite Application Manager Agent for J2EE
   v07.10.01.00
 2) all of the above

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

Type you selections here:
```

9. Type the number corresponding to `all of the above` and press **Enter** and the installer prompts you with the following message to ask you to confirm your selection:

```
The following products will be installed:

 IBM Tivoli Composite Application Manager Agent for J2EE
   V07.10.01.00

Are your selections correct [ 1=Yes, 2=No; default is "1" ]?
```

10. Type **1** and press **Enter** to confirm your selection and start the installation.

11. After installing all of the components, the following message is displayed to ask you whether you want to install components for a different operating system:

    ```
    Do you want to install additional products or product support packages
     [ 1=Yes, 2=No; default is "2" ]?
    ```

    Type **2** and press **Enter**.
12. The installer prompts you with the following message for the configuration:

    ```
    You may now configure any locally intalled IBM Tivoli Monitoring product via
    the "/opt/IBM/ITM/bin/itmcmd config" command.
    ```
13. The installation program will complete the installation and exit. After this, re-configure the desktop client by running:

    ```
    itmcmd config -A cj
    ```

    At any prompts, press **Enter** to accept the default values.

Important: Check that the Eclipse help server has been configured for the client. See "Ensure that the Eclipse server has been configured."

Ensure that the Eclipse server has been configured

After installing application support files on a Tivoli Enterprise Portal Server that provides the browser client or on a Tivoli Enterprise Portal desktop client, you must check the Eclipse help server for the portal client to ensure that it has been configured.

To do this, perform the following procedure:

1. Start Manage Tivoli Enterprise Monitoring Services:

   ```
   ./itmcmd manage
   ```

   The Manage Tivoli Enterprise Monitoring Services window opens.
2. Verify that the Eclipse Help Server entry indicates **Yes** in the Configured column. If it does not, right-click the entry, and select **Configure** from the pop-up menu.
3. You are prompted for the port number that the Eclipse Help Server should use. Verify that this value is set to the same port number you specified when installing IBM Tivoli Monitoring, and click **OK**.

Silent installation and uninstallation

You can use the Installer to install ITCAM Agent for J2EE in silent mode. To do this, modify the sample file provided on the installation DVD, and then run the installer from the command line.

To perform a silent installation, first you need to prepare the response file. Then, run the installer, supplying the name of the response file. A silent uninstallation does not require a response file.

Preparing the response file for the agent installation

To prepare a response file for installing the agent, perform the following procedure:

1. On the product installation image, in the top level directory, locate the `silent_install.txt` file.
2. Make a copy of this file, and open it in a text editor.

3. Modify the following property, if necessary. Do not modify any other properties.

Table 4. Agent installation response file properties

Response file property	Meaning
EncryptionKey	The 32-character encryption key used to secure password transmission and other sensitive data across your IBM Tivoli Monitoring environment. See IBM Tivoli Monitoring: Installation and Setup Guide for details about the encryption key.

4. Save the edited copy in a work directory, for example, as `/tmp/silent.txt`.

Running the Installer in silent mode

After preparing the response file for your installation and uninstallation, run the installer, specifying the path and name for the response file. Perform the following procedure:

1. Change to the directory of the installation image.
2. Invoke `install.sh`:

   ```
   ./install.sh -q -h ITM_home -p response_file_name
   ```

   where *ITM_home* specifies the destination directory where the agent will be installed (by default it is `/opt/IBM/ITM`; you can use different destination directories to install several copies of the agent on the same host); *response_file_name* is the name of the response file you have prepared (with full path). For example:

   ```
   ./install.sh -q -h /opt/IBM/ITM -p /tmp/silent.txt
   ```

Attention: if you are performing an upgrade or maintenance level update, and the Monitoring Agent is currently running, silent installation will be aborted.

Performing a silent uninstallation

To uninstall ITCAM Agent for J2EE in silent mode, perform the following procedure:

1. Change to the *ITM_home*`/bin` directory.
2. Run the command:

   ```
   uninstall.sh -f yj platform_code
   ```

You can find complete information about silent Tivoli monitoring installation in "Appendix B. Performing a silent installation of IBM Tivoli Monitoring" of the *IBM Tivoli Monitoring: Installation and Setup Guide*.

Installing and uninstalling a Language Pack on Linux and UNIX systems

A Language Pack enables user interaction with the agent in a language other than English. For example, when a Spanish language pack is installed, the Tivoli Enterprise Portal workspaces and the internal messages of the Agent are displayed in Spanish.

To enable full support for a language, you must install the Language Pack on the agent host and all hosts where the Tivoli monitoring support files for the agent are

installed (hub Tivoli Enterprise Monitoring Servers, all Tivoli Enterprise Portal Servers, and all Tivoli Enterprise Portal desktop clients).

If you no longer want to use a language, uninstall the language pack for it.

Before installing or uninstalling a Language Pack, ensure that:

- The agent and the Tivoli Enterprise Portal Support Files are installed.
- The Java runtime environment (JRE) is available on every host where you are planning to install the Language Pack. (The JRE is required by IBM Tivoli Monitoring).
- You know the installation directories (*ITM_home*) for the Agent and all other Tivoli monitoring components on which you are planning to install the agent. The default installation directory is `/opt/IBM/ITM`.

Installing a Language Pack on Linux and UNIX systems

To install a Language Pack on Linux and UNIX systems you need to use the installer on the Language Pack DVD. The procedure is the same on the Agent host, hub Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, and Tivoli Enterprise Portal desktop client.

Perform the following procedure:

1. Mount the Language Pack DVD. Make sure the full path to the mount directory does not include spaces.
2. Use the following commands to start the installer from the Language Pack DVD:

   ```
   cd dir_name
   ./lpinstaller.sh -c ITM_home
   ```

3. Select the language of the installer and click OK.

   Note: In this step, you select the language for the installer user interface, not the language pack that will be installed.
4. Click **Next** on the Introduction window.
5. Select **Add/Update** and click **Next**.
6. Select the directory where the the National Language Support package (NLSPackage) files are located. This is the `nlspackage` directory on the Language Pack DVD.
7. Select **ITCAM Agent for J2EE**.
8. Select the languages to install and click **Next**.

   Note: You can hold down the **Ctrl** key for multiple selections.
9. Examine the installation summary page and click **Next** to begin installation.
10. Click **Next**.
11. Click **Finish** to exit the installer.
12. If you are installing the Language Pack on a Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, or Tivoli Enterprise Portal desktop client, start the **Manage Tivoli Monitoring Services** utility, and use it to restart the server or client. If the Eclipse Help Server is running, restart it as well.

Uninstalling a Language Pack on Linux and UNIX systems

To uninstall a Language Pack on Linux and UNIX systems you need to use the installer on the Language Pack image. The procedure is the same on the Agent host, hub Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, and Tivoli Enterprise Portal desktop client.

Perform the following procedure:

1. Mount the Language Pack image. Make sure the full path to the mount directory does not include spaces.
2. Use the following commands to start the installer from the Language Pack image:

   ```
   cd dir_name
   ./lpinstaller.sh -c ITM_home
   ```

3. Select the language of the installer and click OK.

   Note: In this step, you select the language for the installer user interface, not the language pack that will be installed.
4. Click **Next** on the Introduction window.
5. Select **Remove** and click **Next**.
6. Select **ITCAM Agent for J2EE**.
7. Select the languages to uninstall and click **Next**.

   Note: You can hold down the **Ctrl** key for multiple selections.
8. Examine the uninstallation summary page and click **Next** to begin installation.
9. Click **Next**.
10. Click **Finish** to exit the installer.
11. If you are uninstalling the Language Pack on a Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, or Tivoli Enterprise Portal desktop client, start the **Manage Tivoli Monitoring Services** utility, and use it to restart the server or client. If the Eclipse Help Server is running, restart it as well.

Uninstalling ITCAM Agent for J2EE on Linux and UNIX systems

To remove ITCAM Agent for J2EE on UNIX and Linux systems, first unconfigure the Data Collector from all application server instances. For instructions, see "Unconfigure the Data Collector from application server instances" on page 41.

When the Data Collector is unconfigured, perform the following procedure:

1. From a command prompt, run the following command to change to the appropriate /bin directory:

   ```
   cd ITM_home/bin
   ```

2. Run the following command:

   ```
   ./uninstall.sh
   ```

   A numbered list of product codes, architecture codes, version and release numbers, and product titles is displayed for all installed products.
3. Type the number for the monitoring agent. Repeat this step for each additional installed product you want to uninstall.

Chapter 5. Configuring and unconfiguring the monitoring agent and data collector

You must configure the monitoring agent to communicate with the Tivoli Enterprise Monitoring Server and configure the data collector to communicate with the monitoring agent. Also, you must configure the data collector for monitoring every instance of your application server. If you no longer want to monitor an application server instance, unconfigure the data collector for it.

Important: The following application servers are not supported by the current version of the data collector:

- Oracle/BEA application server
- Sun JSAS
- WebSphere Application Server Community Edition

Pre-configuration step: Unconfiguring old data collector

If you are upgrading from an older version of ITCAM Agent for J2EE, you must unconfigure the old data collector before configuring the new data collector.

Use the unconfiguration utility supplied with the old data collector to unconfigure it.

Important: ITCAM Agent for J2EE version 7.1.1 supports only the following application servers:

- WebLogic
- JBoss
- NetWeaver
- Tomcat

It also supports J2SE applications. If your older installation monitors other application servers, do not upgrade it to Agent for J2EE version 7.1.1. Instead, install the latest maintenance levels of the Monitoring Agent and Data Collector for J2EE version 6.2.

Pre-configuration step for monitoring J2SE applications

If there is a startup script for your application, edit the customized script before running the Configuration Tool.

Specify the following two anchors in the customized script to correctly locate the appropriate section of the script and insert the configuration of the Data Collector into the startup script:

- In the startup script, use a new line to define the `ITCAM_DC_SCRIPT` anchor as follows before the execution of the java command:
  - On Windows systems, `REM ITCAM_DC_SCRIPT`
  - On Linux and UNIX systems, `### ITCAM_DC_SCRIPT`

  When the configuration tool runs, it inserts configuration of the Data Collector after this anchor.

- Add the ITCAM_JVM_OPTS anchor as the last JVM option in the customized script. Find where the JVM options are set and insert the anchor:
  - On Windows systems, %ITCAM_JVM_OPTS%
  - On Linux and UNIX systems, ${ITCAM_JVM_OPTS}

  Place the anchor as the last JVM option, before the main class %MAINCLASS% in the J2SE JVM startup options.

Important: Any existing Garbage Collection (GC) logging argument or Java security policy will be overwritten after the configuration process.

Examples:

```
%_EXECJAVA% %JAVA_OPTS% %CATALINA_OPTS% %DEBUG_OPTS%
-Djava.endorsed.dirs="%JAVA_ENDORSED_DIRS%" -classpath "%CLASSPATH%"
-Dcatalina.base="%CATALINA_BASE%" -Dcatalina.home="%CATALINA_HOME%"
-Djava.io.tmpdir="%CATALINA_TMPDIR%" %ITCAM_JVM_OPTS% %MAINCLASS%\
    %CMD_LINE_ARGS% %ACTION%
```

Entering the Agent Configuration window

Use the Agent Configuration window to configure the monitoring agent and the data collector.

Procedure

1. Start the Manage Tivoli Monitoring Services utility:
   - On Windows systems, select **Manage Tivoli Monitoring Services** from the Tivoli Monitoring program folder in the Start menu.
   - On Linux and UNIX systems, change to the *ITM_home*/bin directory (by default, /opt/IBM/ITM/bin) and run the following command:

     ```
     ./itmcmd manage
     ```

     Important: For the Manage Tivoli Monitoring Services utility, the JAVA_HOME environment variable must point to the JVM of the same major version as the JVM used by the application server. If possible, set it to the JVM that the application server uses.
2. Right-click **ITCAM Agent for J2EE** and then click **Configure** or **Reconfigure**.

Results

On Windows systems, the **Agent advanced configuration** window is displayed; see "Configure the monitoring agent connection to the monitoring server" for instructions about using this window. This window does not open at this moment on Linux and UNIX systems.

The agent configuration window opens. Use this window to configure the monitoring agent and data collector.

Configure the monitoring agent connection to the monitoring server

In order to use the agent, you must configure the monitoring agent to communicate with the Tivoli enterprise Monitoring Server.

On Windows systems, after installation of the agent, if you have selected **Configure agents default connection to Tivoli Enterprise Monitoring Server** the **Agent advanced configuration** window opens.

On Windows systems, when you open the agent configuration window (see "Entering the Agent Configuration window" on page 34), the **Agent advanced configuration** window is displayed before you can select the configuration actions.

On Linux and UNIX systems, after you complete a configuration action, the **TEMS Connection** window is displayed.

If IBM Tivoli Monitoring infrastructure is not used (in a deep dive diagnostics only installation), or if you have already set this configuration, ignore this window and click **OK** or **Save**. (Do not click **Cancel**).

Specify the communication protocol and its parameters. Set the host name or IP address of the primary monitoring server and, if available, the secondary monitoring server. If the monitoring agent must access the monitoring server across a firewall, select the **IP.PIPE** protocol and enable the **Connection must pass through firewall** or **Use Address Translation** option.

For more information about communication protocols and their parameters. see *IBM Tivoli Monitoring: Installation and Setup Guide*.

Click **OK** or **Save**.

Configure Monitoring Agent settings

If the IBM Tivoli Monitoring infrastructure is used, you **must** configure Monitoring Agent settings before configuring the Data Collector to monitor any application server instances. Do not perform this configuration in a deep dive diagnostics only installation, where IBM Tivoli Monitoring is not used.

You can change the port that is used for communication between the Data Collector and the monitoring agent (this communication is on the local host); the default port is 63336. You can also set an alternate node name that determines how the agent will be displayed in the Tivoli Enterprise Portal navigation tree.

While you can change these settings at a later time, it is normally most convenient to set them when initially configuring the communication. In this case no manual changes to configuration files is required to change the port number, and no customization of the Tivoli Enterprise Portal view could have been performed by any user. So, if you need to make such changes, make them at installation time if possible.

To configure Monitoring Agent settings, perform the following procedure:

1. Enter the Agent Configuration window. After installation of the Agent on Windows systems, this window opens automatically. Otherwise, see "Entering the Agent Configuration window" on page 34.
2. Select **Configure Tivoli Enterprise Monitoring Agent (TEMA)** and click **Next**.
3. In the Agent Configuration page you can set an alternative Node ID for identifying the agent. This ID, also known as an alias, is the identifier that will determine how the agent will be displayed in the Tivoli Enterprise Portal navigation tree. The default is `Primary`, used in conjunction with the host name of the computer where the Agent is installed. In the **Port** field you can specify a TCP socket port that the monitoring agent will use to listen for connection requests from the Data Collectors. Normally, do not change this value. The port will only be used for local communication on the host. Click **Next**.

Attention: Valid characters for the node ID include A-z, a-z, 0-9, underbar (_), dash (-), and period (.); do not use other characters.

4. The monitoring agent is successfully configured. Click **Home** to return to the **Agent Configuration** window, or click **OK** to complete the configuration process.
5. On Linux and Unix systems, the **Agent advanced configuration** window is displayed; see "Configure the monitoring agent connection to the monitoring server" on page 34 for instructions about using this window.

Configure the Data Collector to monitor application server instances

You must configure the Data Collector for each application server instance that you need to monitor.

To configure the Data Collector to monitor a server instance, perform the following procedure:

1. Enter the Agent Configuration window. After installation of the Agent on Windows systems, if you have selected **Launch Manage Tivoli Monitoring Services for additional configuration options and to start Tivoli Monitoring services**, this window opens automatically. Otherwise, see "Entering the Agent Configuration window" on page 34.
2. Select **Configure Application Server for DC Monitoring** and click **Next**.
3. If you want to configure the Data Collector to communicate with the Managing Server, check the **Enable communication to Managing Server for deep-dive diagnostics** box. Then, Click **Next**. If you left the box unchecked, go to step 7 on page 37.
4. Enter the fully qualified host name of the Managing Server. If a split Managing Server installation is used, this must be the host where the Kernel is located. If the Managing Server is installed on the same host as the Agent, the address and port for this Managing Server will be displayed by default, but you can change them.

   After entering the host name, you can also change the port number on which the Managing Server Kernel is listening. Then, click **Next**.

   Note: This port number is defined as the value of the key "PORT_KERNEL_CODEBASE01" in the .ITCAM61_MS_CONTEXT.properties file located under the Managing Server Home directory. See IBM Tivoli Composite Application Manager for Application Diagnostics Managing Server Installation Guide.
5. Set the Managing Server home directory, which is the destination directory chosen during the installation of the Managing Server.If the Managing Server is running and the configuration utility has been able to communicate to it, its home directory is displayed by default. If the Managing Server is not available at the time of communication, you need to enter the home directory.

   If the Managing Server home directory is not displayed, input it. Click **Next**.
6. If there are multiple IP address on this host, select the address that the Data Collector needs to use for communication with the Managing Server. Also, if you need to change the ports that the Data Collector uses to accept incoming connections from the Managing Server (in case of split Managing Server installation, the Publish Server), select "Specify the RMI Port Number", and enter the "RMI Port Number" and "Controller RMI Port Number". Make sure the ports are not being blocked by the firewall or other applications. The

default RMI port Number range is 8200-8299; the Controller RMI Port Number range is 8300-8399. After making any necessary changes, click **Next**.

7. You can enable the Transaction Tracking API function in the following window. Transaction Tracking Application Programming Interface (TTAPI) enables the integration of ITCAM Agent for J2EE and ITCAM for Transactions. With TTAPI, the Data Collector can send transaction information to ITCAM for Transactions; also, if ITCAM for Application Diagnostics Managing Server is used, transaction specific information is available in the Visualization Engine. TTAPI also enables integration of the Data Collector with the Robotic Response Time component (or T6 agent). To enable TTAPI, check the **Configure Transactions Integration** box, and enter the fully qualified host name or IP address for ITCAM for Transaction Tracking agent and the port number that the Data Collector uses to connect to it. Then, click **Next**. If you do not need to enable the Transaction Tracking API function, leave the box unchecked and click **Next**.
8. A window for choosing the type of application server that the Data Collector monitors is displayed. The following server types are available:
   a. Weblogic Server
   b. SAP NetWeaver Application Server
   c. JBoss Application Server
   d. Tomcat Server
   e. J2SE Application

   Select the application server type, and click **Next**.

The subsequent configuration steps depend on the application server type.

Configuration steps for WebLogic servers

The following steps are specific for WebLogic application servers, including WebLogic Portal Server.

1. Enter the following information about the WebLogic server:
   - WebLogic server home directory
   - Java home directory

     Important: On HP-UX or Solaris systems, select the **Use JDK as 64 bit** check box if you are using a 64-bit JDK.
   - The server host name (the local host is the server host)
   - The port for JMX communication with the server.
   - The username and password for JMX communication with the server.
   - The type of the JNDI protocol: t3 or t3s (one way SSL).
   - If the proticol type is t3s, the path and name of the SSL key file.

   Click **Next**.
2. Select the instances to configure. For every instance, enter the following additional information:
   - The unique alias name for the instance. The name is displayed in the Tivoli Enterprise Portal and, if you have a Managing Server, in te Visualization Engine.
   - If WebLogic is started by a startup script, select **Modify Startup Script File** and select the script file name. The location of the WebLogic startup script is *wl_domain*/bin/startWebLogic.cmd(sh).

Important: If the server instance to be configured is a managed server started by Node Manager, do not select **Modify Startup Script File**.

- If WebLogic is started as a Windows service, select **Installed as Windows service** and enter the service name.

Click **Next**.

For the subsequent configuration steps, see "Final configuration steps" on page 40.

Configuration steps for NetWeaver servers

The following steps are specific for NetWeaver application servers.

1. Enter the NetWeaver server information. Select the NetWeaver server version and the installation type. For a detailed description of the available the installation types, see "Three installation types of ITCAM for J2EE Data Collector for NetWeaver" on page 8. Other settings in this window depend on the installation type:
   - In a central instance installation, the central instance is monitored. Select the **Server Home** directory, which is the absolute path of the central instance home directory (for example, C:\usr\sap\J2E\JC00).
   - In a local dialog instance installation, the central instance and the dialog instance are on the same host and the dialog instance is monitored. Select the following directory paths:
     - The **Server Home** directory is the absolute path of the local dialog instance home directory (for example, C:\usr\sap\J2E\J01).
     - The **Central Instance Home** directory is the absolute path of the central instance home directory (for example, C:\usr\sap\J2E\JC00).
   - In a distributed dialog instance installation, the central instance and the dialog instance are on different hosts and the dialog instance is monitored. Select the following directory paths:
     - The **Server Home** directory is the absolute path of the distributed dialog instance home directory on the local host (for example, C:\usr\sap\J2E\J01).
     - The **Central Instance Home** directory is the absolute path of the central instance home directory on the host where the central instance is running (for example, C:\usr\sap\J2E\JC00).
     - The **Central Instance Network Home** directory is the path of the central instance home directory, as mounted from the local host (for example, Y:\usr\sap\J2E\JC00). Ensure that the user account used for configuring and running the Data Collector can access this directory.

   Also, enter the following information:
   - The Java home directory that the application server uses.
   - The host name for the NetWeaver host (the local host). The configuration utility automatically determines the host name and displays it in this field.
   - The port number for communication with this server.
   - The protocol type for communication with this server (P4, SSL, or HTTP)
   - The user name and password for communication with this server, created during your installation of NetWeaver. Usually you use these user name and password to log on to the Visual Administrator tool.

   Click **Next**.
2. Select the server instance that is to be monitored. Click **Next**.

For the subsequent configuration steps, see "Final configuration steps" on page 40.

Configuration steps for JBoss servers

The following steps are specific for JBoss application servers.

1. Select the JBoss server home directory, the home directory of the Java virtual machine that the JBoss server uses, and the JBoss startup file. Click **Next**.
2. Review the server information. Click **Next**.
3. Select the server instances to monitor. For each selected instance, enter the alias name that will be displayed in Tivoli Entreprise Portal and, if you have the Managing Server, in the Visualization Engine. Click **Next**.

For the subsequent configuration steps, see "Final configuration steps" on page 40.

Configuration steps for Tomcat servers

The following steps are specific for Tomcat application servers.

Enter the following information:

- The Tomcat server home directory
- The Java home directory

  Important: On HP-UX or Solaris systems, select the **Use JDK as 64 bit** check box if you are using a 64-bit JDK.
- The way Tomcat starts up: **Normal Startup** (with a startup script), **Using Java Wrapper**, or **Using Windows Service**
- For **Normal Startup** select the Tomcat startup script file. For **Using Java Wrapper**, select the Java Service Wrapper startup file and configuration file. For **Using Windows Service**, enter the Windows service name.

Click **Next**.

For the subsequent configuration steps, see "Final configuration steps" on page 40.

Important: To support cascading configuration files in the java service wrapper framework, the base directory of the java service wrapper needs to be specified. This enables a user to define a relative path for an included configuration file. By default, the base-directory is the location of the wrapper.exe in Windows, or the script used to launch the wrapper in Unix. If the `wrapper.working.dir` property is defined in the java service wrapper configuration file, ITCAM will use the value of the property as the base directory.

The java service wrapper configuration file is similar to the java properties file. It contains the information necessary to launch a JVM instance with the correct command line required by an application. The default file is wrapper.conf. When you configure Tomcat to use the java service wrapper, ITCAM creates a new file called itcam_wrapper.conf in the same directory as wrapper.conf. This file includes all ITCAM configuration items. The wrapper.conf file references the itcam_wrapper.conf file using an include statement.

Configuration steps for a J2SE application

The following steps are specific for monitoring a J2SE application.

1. Enter the J2SE application information:
   - In the **Server Home** field, click **Browse** to locate the directory in which J2SE has been installed. Ensure the J2SE server home value you enter is correct as there is no validation performed on this field. The value will be displayed in Tivoli Enterprise Portal after the installation.

- For the **Java Home** field, click **Browse** to locate the JDK that is supporting the application.

  Important: On HP-UX or Solaris systems, select the **Use JDK as 64 bit** check box if you are using a 64-bit JDK.
- For the **Main Class** field, locate the .bat file under the J2SE server home directory and copy the fully qualified main class name from the Java command line. For example, `com.sample.MyApp` from the following line:

  ```
  java $ITCAM_JVM_OPTS -classpath $CLASSPATH com.sample.MyApp hello
  ```
- Select **JMX Server Remotely connect** if the JMX server requires an RMI connection even from the local host.

Click **Next**.

2. Review the information gathered from the specified J2SE application and click **Next**.
3. Enter the information required to configure the J2SE instance:
   - In the **Instance name** field, enter the name for this application to be displayed in the TEP and, if a Managing Server is used, in the VE. Use only English and numeric characters.
   - In the **JVM Arguments** field, enter the JVM options without the main class name, for example: -Dvarible1=value1
   - In the **Program Arguments** field, enter the program arguments after the main class.
   - Select the **Modify existing startup script file** box to modify the current application startup script to start the Data Collector. If you clear the box, a sample script is created instead, and you will need to change the startup script later based on the generated sample script.
   - If you select **Modify existing startup script file**, select the script file. Otherwise, select the path for the sample script.

     Important: The **JVM Arguments** and **Program Arguments** fields are used only if you create a sample script file. If you modify the startup script file, the configuration utility ignores these values.

   Click **Next**.

For the subsequent configuration steps, see "Final configuration steps."

Final configuration steps

Complete the following steps at the end of the configuration, after the steps specific for an application server.

1. The configuration utility validates the application server connection and applies the configuration. Click **Next**
2. The configuration summary information is displayed. To export the summary report to a file, click the **Export Summary Report** button. The application server needs to be restarted before the Data Collector configuration takes effect. Click **Home** to return to the **Agent Configuration** window, or click **OK** to complete the configuration process.

   Important: After configuring the Data Collector to monitor an application server instance, perform the applicable post-configuration steps, including a restart of the application server. The Data Collector configuration will take effect after the server is restarted.

3. On Linux and Unix systems, the **Agent advanced configuration** window is displayed; see "Configure the monitoring agent connection to the monitoring server" on page 34 for instructions about using this window.

Unconfigure the Data Collector from application server instances

If you no longer want to monitor an application server instance, unconfigure the Data Collector from it. If you want to uninstall ITCAM Agent for J2EE, you must unconfigure the Data Collector from all instances.

To unconfigure the Data Collector from server instances, copmlete the following procedure:

1. Enter the Agent Configuration window. After installation of the Agent on Windows systems, if you have selected **Launch Manage Tivoli Monitoring Services for additional configuration options and to start Tivoli Monitoring services**, this window opens automatically. Otherwise, see "Entering the Agent Configuration window" on page 34.
2. Select **Unconfigure Application Server for DC Monitoring** and click **Next**.
3. Select the instances to unconfigure and click **Next**.
4. The configuration utility validates the application server connection and applies the unconfiguration. Click **Next**
5. The unconfiguration summary information is displayed. To export the summary report to a file, click the **Export Summary Report** button. The application server needs to be restarted before the Data Collector unconfiguration takes effect. Click **Home** to return to the **Agent Configuration** window, or click **OK** to complete the configuration process.
6. On Linux and Unix systems, the **Agent advanced configuration** window is displayed; see "Configure the monitoring agent connection to the monitoring server" on page 34 for instructions about using this window.

Completing a silent configuration

You can use the Configuration utility in Silent mode to perform all configuration tasks (including unconfiguration) for ITCAM Agent for J2EE. To do this, prepare the response file by modifying a sample provided with the Agent.

All configuration tasks for the Agent can also be performed in Silent mode, without user interaction. This may be especially useful for large-scale deployments.

To perform a configuration task, you need to prepare a response file, and then start the configuration utility.

Preparing a response file

To perform a configuration task using silent mode, create a copy of a sample response file for the task. Modify this copy, and save it in a work directory, for example, as `C:\TEMP\SILENT`.

For each of the configuration tasks for the agent, a sample response file is available in the *ITM_home*`\TMAITM6` directory (on Windows systems) or in the *ITM_home*`/samples` directory (on Linux and UNIX systems). Make a copy of the file and edit it as required, using the information provided in the comments within the file.

- **Configuring Monitoring Agent connection to the Monitoring Server and Data Collector connection to the monitoring agent**, while two separate tasks in the GUI configuration (see "Configure the monitoring agent connection to the monitoring server" on page 34 and "Configure Monitoring Agent settings" on page 35), are performed with one response file. If the Agent is to communicate with the IBM Tivoli Monitoring infrastructure, you must perform this configuration task before configuring the Data Collector to monitor any application server instances. Do not perform this task if Tivoli Monitoring is not used (in a deep dive diagnostics only installation). The sample file name is `yjv_silent_config_agent.txt`.
- **Configuring the Data Collector to monitor an application server instance**: the sample file names for the different application servers are:
  - For J2SE, `yjv_silent_config_j2sedc.txt`
  - For JBoss, `yjv_silent_config_jbossdc.txt`
  - For NetWeaver, `yjv_silent_config_netweaverdc.txt`
  - For Tomcat, `yjv_silent_config_tomcatdc.txt`
  - For WebLogic, `yjv_silent_config_wlsdc.txt`
- **Unconfiguring the Data Collector from an application server instance**: the sample file names for the different application servers are:
  - For J2SE, `yjv_silent_unconfig_j2sedc.txt`
  - For JBoss, `yjv_silent_unconfig_jbossdc.txt`
  - For NetWeaver, `yjv_silent_unconfig_netweaverdc.txt`
  - For Tomcat, `yjv_silent_unconfig_tomcatdc.txt`
  - For WebLogic, `yjv_silent_unconfig_wlsdc.txt`

The response file is a text file, containing parameter names and values in the format *parameter=value*, for example:

```
KERNEL_HOST01=servername.domain.com
```

Comment lines begin with a number sign (#). Do not use blank lines.

Any \ character must be escaped as \\, : as \:, and spaces must be prefixed with \, for example:

```
MS_AM_HOME=C\:\\Program\ Files\\ITCAM\\MS
```

In the file sections marked as "repeatable", parameters are specific to a profile path or an application server instance name. For these parameters, use the path or name as a key, in the format *parameter.key=value*. For example:

```
wls-ALIAS_INSTNAME.INST1=myserver
wls-CUSTOM_SCRIPT_ENABLED.INST1=true
```

Use the actual values instead of parameters that are market by <> brackets. For example, replace `<NetWeaver INSTANCE NAME>` with the actual NetWeaver instance name.

Running the Configuration utility in silent mode

After preparing the response file for a configuration task, run the configuration utility, specifying the path and name for the response file.

On Windows systems complete the following procedure:

1. Open a Windows command prompt window, and change to the *ITM_home*\installITM directory.

2. Invoke the configuration utility as follows. Specify the parameters in the exact order shown:

   ```
   kinconfg -nresponse_file_name -ckyj
   ```

   where *response_file_name* is the name of the response file you have prepared (with full path). For example:

   ```
   kinconfg -nC:\TEMP\SILENT.TXT -ckyj
   ```

On Linux and UNIX systems complete the following procedure:

1. Change to the *ITM_home*/bin directory.
2. Invoke the configuration utility as follows. Specify the parameters in the exact order shown:

   ```
   ./itmcmd config -A -p response_file_name yj
   ```

   where *response_file_name* is the name of the response file you have prepared (with full path). For example:

   ```
   ./itmcmd config -A -p /tmp/silent.txt yj
   ```

Chapter 6. Post-configuration tasks

Depending on your application server type, you must complete certain tasks after configuring ITCAM Agent for J2EE to monitor the server.

Post-configuration steps for ITCAM for J2EE Data Collector

1. Increase the JVM Maximum Heap Size by at least 128 megabytes.

2. Apply the latest level of maintenance (such as fix packs or interim fixes) from the following Web site:

http://www-947.ibm.com/support/entry/portal/product/tivoli/tivoli_composite_application_manager/tivoli_composite_application_manager_for_j2ee

Post-configuration steps for all application servers using Sun JDK 1.5 or HP JDK 1.5

This applies only if you have installed the Java Virtual Machine Tool Interface (JVMTI) interim fix.

If your application server is using Sun JDK 1.5 (J2EE or Community Edition application servers) or HP JDK 1.5 (J2EE application servers only), you need to set the JVM parameter MaxPermSize to -XX:MaxPermSize=196m or above in order to prevent out-of-memory errors.

Post-configuration steps for all application servers using Sun JDK

For Sun JDKs, Data Collector configuration enables verbose garbage collection output by -Xloggc JVM argument. By default, the -Xloggc causes JVM to generate class loading and unloading events to native standard output stream, if user chooses to redirect it to log files, it may fill the log files and consume excessive disk space.

To suppress class loading and unloading events, add the *-XX:-TraceClassUnloading -XX:-TraceClassLoading* options to the JVM argument of the application server. Please refer to the administration guide of the application server for instructions on how to add options to JVM arguments.

For more information about the -XX:-TraceClassUnloading -XX:-TraceClassLoading options, refer to:

http://java.sun.com/docs/hotspot/gc1.4.2/faq.html

http://java.sun.com/docs/hotspot/gc5.0/gc_tuning_5.html

Post-configuration steps for Tomcat users

You need to perform some post-configuration steps if your Tomcat server is started by Java Service Wrapper. If you want to reconfigure the DC right after it is unconfigured, continue your reconfiguration and the DC configuration tool will pick up all properties in the itcam_wrapper.conf file and reuse them. If you want to change the wrapper.conf file after the DC is unconfigured, perform this procedure:

1. Manually remove the whole ITCAM Configuration section which begins with the line `###include ITCAM Data Collector Configuration File Begin` and ends with the line `###include ITCAM Data Collector Configuration File End` in the `wrapper.conf` file.
2. If there are missing numbers in JVM arguments after you removed the section above, please follow the Java Service Wrapper guidelines and add the missing properties or change the numbering of other properties to ensure that the `wrapper.conf` file is well formed.
3. Permanently remove the `itcam_wrapper.conf` file from disk.
4. At this time, ITCAM will be completely unconfigured and you can continue your changes on the `wrapper.conf` file.

Post-configuration steps for WebLogic users

The following post-configuration steps are specific for WebLogic users.

Restarting and shutting down the application server

Restart your application server to enable the configuration and make sure to shut down the WebLogic application server instance through the Administration Console. For more detailed information about how to shut down the WebLogic application server, refer to the following Web site:

http://docs.oracle.com/cd/E13222_01/wls/docs90/server_start/startquickref.html.

If the configured application server instance is controlled by a Node Manager, restart the Node Manager as well.

If a application server instance in which the Data Collector is configured is an administrative application server instance, some exceptions are produced when the application server is shutting down. Ignore these exceptions.

For users who start the managed server from the Node Manager, the following JVM property must be added:

```
-Dcom.ibm.tivoli.jiti.injector.IProbeInjectorManager=com.ibm.tivoli.itcam.toolkit.
ai.bcm.bootstrap.ProbeInjectorManager
```

Refreshing the Windows service

On Windows, if WebLogic is installed as a Windows service, you need to refresh the service. The procedure depends on whether WebLogic is started by Node Manager.

If WebLogic is running in Windows service mode, not started by Node Manager:

1. After the Data Collector is configured successfully, if the WebLogic Windows service is running, stop it and run `uninstallService.cmd` .

2. Reinstall the Windows service by using the following command: `InstallService.cmd` *user_id user_pwd* .
3. Open the system service window and start the WebLogic server.
4. If any problems occur, find the cache file in directory *domain_dir**instance_dir**.wlnotdelete**extract* and remove the following directories:

   *instance_name*_console_console

   *instance_name*_uddi_uddi

   *instance_name*_uddiexplorer_uddiexplorer

   *instance_name*_wl_management_internal1_wl_management_internal1

   *instance_name*_wl_management_internal2_wl_management_internal2

domain_dir refers to the name of the domain, and *instance_name* refers to the server instance name. For example, if you create a basic portal domain named *portalDomain*, and a server instance named *portalServer*, the cache files would be found in the \portalDomain\portalServer\.wlnotdelete\extract directory.

If WebLogic is running in Node Manager Windows service mode:

1. After the Data Collector is configured successfully, if the Node Manager server service is running, stop it and run `uninstallNodeMgrSvc.cmd` .
2. In the directory *AppServer_home*/server/bin, run `installNodeMgrSvc.cmd` .
3. Open the system service window and start the Node Manager service.

Use Node Manager to start the managed server. You do not have to run `startNodeManager.cmd` .

Post-configuration steps for J2SE applications

JMX server settings

If there is a custom JMX implementation for the application, write an implementation class to implement JMXEnginePlugin interface. This class must implement the JMXEnginePlugin interface, which is described in "J2SE JMXEnginePlugin interface" on page 79.

In "J2SE JMX plug-in sample" on page 80, there is a sample java file, which shows you how to implement the JMXEnginePlugin interface.

To enable this class, set it in the classpath, and edit <*DC_Home*>/runtime/<*server_type*>.<*app_name*>.<*host_name*>.<*inst_name*>/datacollector.properties file, set j2se.jmxe.pluginclass as the custom class name.

If your JDK version is 1.5 and there is no default JMX implementation for the application, append the following information in the startup script of J2SE:

- For users of SUN JRE, append `set ITCAM_JVM_OPTS=%ITCAM_JVM_OPTS% -Dcom.sun.management.jmxremote` to the startup script to enable your remote management.
- For users of IBM JRE, append `set ITCAM_JVM_OPTS=%ITCAM_JVM_OPTS% -Dcom.sun.management.jmxremote.port=0 -Dcom.sun.management.jmxremote.ssl=false -Dcom.sun.management.jmxremote.authenticate=false`.
- For users of BEA JRE, append `set ITCAM_JVM_OPTS=%ITCAM_JVM_OPTS% -Xmanagement`.
-

If you are not using a JDK with version 1.5 or there is a default JMX implementation for the application, ignore this message.

Enabling special request monitoring

To enable the data collector to monitor JDO/CTG/MQI/JMX requests, edit the `toolkit_custom.properties` file in the *DC_home*`\runtime\`*appname*`.`*instname*`.`*hostname*`.`*dcname*`\custom` directory. The DC_home directory is

If you want to monitor CTG requests, set `-Dam.sdc.probe.l1aspectfamily.ctg=CTGASPECTS,both` in your **Java Options**.

If you want to monitor Remote Method Invocation over Internet InterORB Protocol (RMI/IIOP) requests, set

```
-Dorg.omg.PortableInterceptor.ORBInitializerClass.com.ibm.tivoli.itcam.
dc.orbinterpretor.Initializer
```

in your **Java Options**.

Post-configuration steps for NetWeaver

Configuring NetWeaver to monitor system resources

You must make some configuration changes in Netweaver in order to have data reported in System Resources. To enable the system resources monitoring, perform the following steps:

1. Logon the Visual Administrator.
2. Select the target server and then *Services -> Monitoring -> Root*
3. Subnodes are shown after expanding the tree node *Root* and each node attribute represents one type of system resources metric. Please use the following mapping for enabling the metrics displayed in System Resources. After modifying the attribute, go to *Monitoring Configuration* panel, select *Configuration -> Edit -> Save* to save the changes.

Table 5. Metrics displayed in System Resources

Metric in System Resources	**Node attribute in Netweaver**
Component Performance	Services -> Monitoring -> Root -> Performance -> Application Response Time -> Component Content
Request Performance	Services -> Monitoring -> Root -> Performance -> Application Response Time -> Request Content
Performance Summary	Services -> Monitoring -> Root -> Performance -> Application Response Time -> Summary Content
Thread Pool	Services -> Monitoring -> Root -> Kernel -> Application Threads Pool -> * Services -> Monitoring -> Root -> Kernel -> System Threads Pool -> *
Web Container	Services -> Monitoring -> Root -> Services -> Web Container -> *
Entity EJB	Services -> Monitoring -> Root -> Services -> EJB -> Entity Beans -> ** -> *Bean -> *
Stateless EJB	Services -> Monitoring -> Root -> Services -> EJB -> Session Stateless Beans -> ** -> *Bean -> *

Table 5. Metrics displayed in System Resources (continued)

Metric in System Resources	Node attribute in Netweaver
Stateful EJB	Services -> Monitoring -> Root -> Services -> EJB -> Session Stateful Beans -> ** -> *Bean -> *
Message EJB	Services -> Monitoring -> Root -> Services -> EJB -> Message Driven Beans -> ** -> *Bean -> *
Transaction	Services -> Monitoring -> Root -> Services -> Transactions -> *
Memory	Services -> Monitoring -> Root -> Services -> Memory -> *
JVM	Services -> Monitoring -> Root -> System -> VM info -> *
System	Services -> Monitoring -> Root -> System -> System Properties -> *
Web Service Performance	Services -> Web Services -> Performance Data -> ** -> Implementation Time/PostProcessing Time/Preprocessing Time
Web Service Request	Services -> Web Services -> Requests Number -> ** -> CurrentClient/FailedRequests/SuccessfulRequests
HTTP	Services -> Monitoring -> Root -> Services -> Http Provider -> **

Configuring NetWeaver to monitor the HTTP session

Configure the HTTP session settings in NetWeaver to obtain live session data in Server Overview and Server Activity Display in the Application Monitor user interface. To configure the HTTP session settings in NetWeaver, perform the following steps:

1. Log in to the Visual Administrator tools.
2. Go to **Server instance > services > monitoring > Services > Web Container > CurrentHttpSessions**.
3. In the **Monitoring Configuration** panel, click **Configuration**.
4. Edit the current HTTP session settings and save the settings.

Import the JVM parameters of DC for NetWeaver to monitor the server on the distributed dialog instance

This step is only required when you select the installation type as **Distributed dialog instance installation**.

If the ITCAM for J2EE DC is installed on the distributed dialog instance computer, manually configure your DC on central instance computer. Complete this task after the data collector configuration is finished. Configure the following steps before "Configuring references from J2EE services to Tivoli custom service" on page 50.

1. Log on the central instance computer.
2. Navigate to <*Central instance home*>/j2ee/configtool
3. Edit BatchConfig.bat on Windows platform or BatchConfig.sh on UNIX/Linux platforms. Modify the <*Java home*> setting as the <*Java home*> used by the central instance.
4. Navigate to <*Central instance home*>/SDM/program
5. Run config.bat on Windows platforms or config.sh on UNIX/Linux platforms. For unconfiguration, run unconfig.bat or unconfig.sh.

Note: Before you perform the post configuration steps, it is recommended to save a backup of the database of NetWeaver J2EE Engine.

Configuring references from J2EE services to Tivoli custom service

You need to set up 6 references of the J2EE services in the NetWeaver server to the Tivoli custom service. The services to be modified are shown in the following table:

Table 6. Services and related xml files

Service name	Related xml file
connector	connector-provider.xml
naming	naming-provider.xml
servlet_jsp	servlet_jsp-provider.xml
ejb	ejb-provider.xml
jms_provider	jms_provider-provider.xml
jmsconnector	jmsconnector-provider.xml

Apply the following steps to setup the references one by one:

1. Start the J2EE Engine Visual Administrator and connect it to the J2EE Engine.
2. Select **Server** > **Services** > **Configuration Adapter Service**.
3. Select **Runtime** > **Display Configuration**.
4. Select **Edit mode**.
5. Select **cluster_data** > **server/dispatcher** > **cfg** > **services** > <*component_name*>-provider.xml. In the dialog window that is displayed, add the component reference into the configuration of components respectively:

   ```
   <reference type="service" strength="weak">
       tivoli
   </reference>
   ```

   .
6. Click **OK** to save your changes.

   Note: You need to repeat steps 5 and 6 to set up the references for all the components.
7. Restart the corresponding cluster element.

CAUTION:
The Tivoli service is not undeployed during unconfiguration. You cannot undeploy it because all the Data Collectors share the Tivoli service. If you want to undeploy the Tivoli service, complete the following steps before undeployment.

- **Unconfigure all Data Collectors from all servers on the corresponding instance.**
- **Remove references from servlet_jsp, naming, ejb, jms_provider, jms_connector, and connector components to Tivoli component. Remove the bidirectional references between the CTG/JDO/IMS/MQI library components and Tivoli service component.**

Otherwise, the Netweaver server cannot start.

For more information about how to modify the reference of a component, refer to the *SAP Note (857025)*.

Configuring ITCAM for J2EE DC for NetWeaver to monitor the CTG/JDO/MQI/IMS

When CICS® Transaction gateway (CTG), Java Data Objects (JDO), Message Queue Interface (MQI), or IMS™ are deployed as libraries, to monitor their request data, perform the following configuration steps.

Make sure that there are bidirectional references between Tivoli service component and CTG/JDO/MQI/IMS library component. For example, if the CTG jars is deployed as the CTGLIB Library, complete the following steps:

1. Start the J2EE Engine Visual Administrator and connect it to the J2EE Engine.
2. Select **Server** > **Services** > **Configuration Adapter Service**.
3. Select **Runtime** > **Display Configuration**.
4. Select **Edit mode**.
5. Select **cluster_data** > **server/dispatcher** > **cfg** > **services** > <*component_name*>-provider.xml. In the dialog window that is displayed, add the component reference before </references>:

   ```
   <reference type="service" strength="weak">
        tivoli
    </reference>
   ```

   .
6. Select **cluster_data** > **server/dispatcher** > **cfg** > **ext** > tivoli-provider.xml. In the dialog window that is displayed, add the component reference before </references>:

   ```
   <reference type="library" strength="weak">
        CTGLIB
    </reference>
   ```

   .
7. Click **OK** to save your changes.

   Note: If you want to monitor JDO, MQI, and IMS, repeat steps 5 and 6. Establish bidirectional references between JDO, MQI, or IMS library component and the Tivoli service component.
8. Restart the corresponding cluster element.

CAUTION:
If CTG/JDO/IMS/MQI deployed as libraries in the NetWeaver server and the DC is installed to monitor the server, and you want to undeploy CTG/JDO/IMS/MQI library components, remove the bidirectional references between Tivoli service component and the library components to be undeployed. Otherwise, the Netweaver server cannot start.

For more information about how to modify the references of a component, refer to the *SAP Note (857025)*.

Additional post-configuration tasks

Perform the following steps:

1. Restart the instance of the application server that will be monitored by the Data Collector.

   If the application server fails to start up, Data Collector configuration has failed. See 2 on page 52.

2. You know the Data Collector configuration has failed if any of the following has occurred:
   - After the configuration, the application server fails to restart.
   - During a GUI configuration, the summary panel for the Configuration Tool indicates the configuration has failed.
   - During a silent configuration, the command line indicates a message that the configuration has failed.
   - After the configuration, there are messages in the Tivoli common log file that indicates configuration has failed.
3. Perform the tasks described in each of the following sections, if applicable.
4. If Terminal Services is enabled on **Windows 2000** or **Windows 2003 Server**, run the following from a command prompt:

   ```
   change user /execute
   ```

Enabling instrumentation and monitoring of RMI/IIOP requests between two application servers

If two application servers are using Remote Method Invocation over Internet InterORB Protocol (RMI/IIOP), and you need to enable instrumentation and monitoring of RMI/IIOP requests and view correlation icons in the Application Monitor user interface, both servers must be instrumented by Data Collectors connected to the same Managing Server. Also, for both application servers, you must set an additional JVM parameter.

On each of the hosts, use your Application Server to add the following parameter for the Java Virtual Machine:

```
-Dorg.omg.PortableInterceptor.ORBInitializerClass.com.ibm.tivoli.itcam.dc.
  orbinterpretor.Initializer
```

More than one Data Collector installed on a server with a firewall enabled: setting a range of port numbers

If you configure communication with the Managing Server, he configuration program requires you to set unique port numbers for probe.rmi.port and probe.controller.rmi.port. Communication problems with the Managing Server arise if ports for separate Data Collectors installed on a server with a firewall are not unique. If you have many Data Collectors, it might be difficult to set unique ports for all the Data Collectors.

Instead of ensuring that individual port numbers assigned for each of the Data Collectors are unique, you can set a range of port numbers in the Data Collector properties file, `datacollector_custom.properties`. See “Fine-tuning the data collector properties files” on page 55 for the location of the file when monitoring different application servers.

The following procedure resets the individual port numbers entered during the configuration to a range of port numbers:

For each `datacollector_custom.properties` file, set the following properties:

```
probe.rmi.port=range_of_port_numbers
probe.controller.rmi.port=range_of_port_numbers
```

For example,

```
probe.rmi.port=8200-8299
probe.controller.rmi.port=8300-8399
```

If you use the same range for both properties, make sure that range is larger than or equal to twice the number of Data Collectors installed on the server.

Linux and UNIX systems: If you used the root ID for the agent installation and the application server is not owned and operated by the root ID

On Linux and UNIX systems, you might use the root user ID to perform the agent installation. The installer will have the authority to use whatever directories and files it requires, and will be able to find most application server installations on the server. But, if the application server is not owned and operated by root ID, you will need to make the following change in order for the Data Collector to work correctly:

- Use the chown command to turn over the Data Collector installation from root to the application server owner ID:

  ```
  chown -R serverOwnerId:serverGroupId DC_home
  ```

Restarting the application server

After completing configuration of the Data Collector, you must restart the monitored application server.

Chapter 7. Customization and advanced configuration for the Data Collector

This section contains instructions for customizing your configuration of the Data Collector (DC).

Fine-tuning the data collector properties files

To best suit the needs of your environment, you can fine-tune the settings in the Data Collector properties and toolkit properties files. The files are specific to each monitored server instance (or J2EE application) and are located under the following instance directory, depending on the application server:

Table 7. Locations of the Data Collector instance directory

WebLogic	If the monitored server instance is represented by a weblogic machine: *DC_home*/runtime/wls*app_server_version.domain_name.machine_name.instance_name*/datacollector.properties else: *DC_home*/runtime/wls*app_server_version.domain_name.host_name.instance_name*.datacollector.properties
Tomcat	*DC_home*/runtime/tomcat*app_server_version.host_name.instance_name*/*DC_home*/runtime/datacollector.properties
JBoss	*DC_home*/runtime/jboss*app_server_version.host_name.instance_name*/jboss*app_server_version.host_name.instance_name*.datacollector.properties
NetWeaver	*DC_home*/runtime/netweaver*app_server_version.sap_node_ID_host_name.sap_instance_number*/datacollector.properties
J2SE	*DC_home*/runtime/j2se.*application_name.host_name.instance_name*/*DC_home*/runtime/datacollector.properties

The Data Collector properties file

The data collector properties file is automatically created by the data collector, and is unique for every application server instance that is monitored by the data collector. It is located in the instance directory and its name is `datacollector.properties`.

However, to facilitate future upgrades, do not change this file.

Instead, add the settings that you want to modify to the data collector custom properties file. This file is located in the `custom` subdirectory of the instance directory. Its name is `datacollector_custom.properties`.

Important: If the `datacollector_custom.properties` file does not exist, create it when you want to make changes. You might also have to create the `custom` subdirectory in the instance directory.

The following properties are in the Data Collector properties file. Only the properties that are recommended for you to modify are listed.

kernel.codebase
The value of this property is filled in during installation time by the installer. It specifies where the Managing Server codebase can be found.

kernel.rfs.address
The value of this property is filled in during installation time by the installer. This value is used by the Application Monitor to locate the Managing Server components.

probe.library.name
The default value is am. This property specifies the name of the native shared library which the Data Collector needs to run. If the value of the property is am, the Data Collector searches for a shared library. This shared library is named libam.so on UNIX platforms and libam.dll on the Windows platform. In normal cases, this property does not need to be specified or changed from the default. Only when the user needs to run a native shared library with a different name does this property need to change.

Example:

```
probe.library.name=am
```

internal.probe.event.packet.size
The default value is 70 or (70 X 1024 kbytes). Changing to below the default is not recommended. Valid values are 1 - 4000000 (or up to available process memory on the server). This property specifies the size of the Data Collector's internal send buffer. The send buffer controls how much data the Data Collector can be sent to the Publish Server at a given time. In normal situations, this property does not have to be changed, as the default send buffer size is more than adequate. However, if the user sees a problem with the amount of data the Data Collector sends to the Publish Server, this property can be set to configure the size of the send buffer.

internal.memory.limit
The default value is 100 (MB). This property limits the amount of memory the Data Collector can use.

internal.memory.accept.threshold
The default value is 2 (MB). This property specifies the minimum free memory after which the Data Collector starts accepting data once it reaches the upper limit. The upper limit is specified by the internal.memory.limit property.

internal.url.limit
The default value is 1000. This property controls the maximum URL length accepted by the Data Collector.

internal.sql.limit
The default value is 5000. This property controls the maximum SQL length accepted by the Data Collector.

internal.probe.event.queue.size.limit
The default value is 900000. This property controls the maximum size of the queue of events maintained by the Data Collector. When the queue is full, the Data Collector drops events.

internal.lockanalysis.collect.L*n*.lock.event
The variable *n* can represent Mod L1, L2, or L3. Possible values are true or false. This parameter controls whether lock acquisition/release events are

collected. The recommended setting at all levels is false as there is little benefit in displaying lock acquisition events if they are not experiencing contention.

Example:

```
internal.lockanalysis.collect.L1.lock.event = false
```

internal.lockanalysis.collect.L*n*.contend.events

The variable *n* can represent Mod L1, L2, or L3. Possible values are true, false, or justone. This parameter controls whether lock contention events are collected.

True indicates contention records are collected. For each lock acquisition request that results in contention, a pair of contention records is written. These records are written for each thread that acquired the lock ahead of the requesting thread. False indicates contention records are not written. Justone indicates contention records are written. However, a maximum of one pair of contention records are written for each lock acquisition request that encounters contention. This event occurs regardless of how many threads actually acquired the lock prior to the requesting thread.

Setting this parameter to true enables you to determine the problem. You can check if a single thread is holding a lock for an excessive time, or if the problem is due to too many threads all attempting to acquire the same lock simultaneously. The recommended setting at L1 is false. The recommended setting at L2 is justone. This setting enables you to collect just one pair of contention records for each lock acquisition that encountered contention. The recommended setting at L3 is true but for a limited time to reduce performance cost. This setting enables you to identify every thread that acquired the lock ahead of the requesting thread.

Example:

```
internal.lockanalysis.collect.L2.contend.events = justone
```

internal.lockanalysis.collect.L*n*.contention.inflight.reports

The variable *n* can represent Mod L1, L2, or L3. Possible values are true or false. This parameter controls whether data is collected for the Lock Contention report. The recommended setting at L1 is false. The recommended setting at L2 and L3 is true.

Example:

```
internal.lockanalysis.collect.L3.contention.inflight.reports = true
```

deploymentmgr.rmi.port

It is not necessary to define the property deploymentmgr.rmi.port if you are running a stand-alone application server. This property is needed for version 5 application server clusters or application servers controlled by a Deployment Manager.

Example:

```
deploymentmgr.rmi.port=<Deployment Manager RMI (bootstrap) port>
```

deploymentmgr.rmi.host

It is not necessary to define the property deploymentmgr.rmi.host if you are running a standalone application server. This property is needed for version 5 application server clusters or application servers controlled by a deployment manager.

Example:

```
deploymentmgr.rmi.host=<Deployment Manager host>
```

networkagent.socket.resettime

The default is no reset. Time interval after which the connection between the Data Collector and the Publish Server is reset.

Example:

```
networkagent.socket.resettime=-1
```

am.mp.cpuThreshold

The default is 30 milliseconds. Only the methods which take at least the minimum amount of CPU time specified in this property are captured for method profiling data. This property avoids unnecessary clutter. Generally, methods with greater than the value specified in this property are considered useful. Customers can reduce or increase this value if needed.

am.mp.clockThreshold

The default is 30 milliseconds. Only the methods which take at least the minimum amount of wall clock time specified in this property are captured for method profiling data. This property avoids unnecessary clutter. Generally, methods with greater than the value specified in this property are considered useful. Customers can reduce or increase this value if needed.

am.mp.leagueTableSize

The default is 1000. This value is the maximum number of methods that are monitored for method profiling data. Customers can reduce or increase this value if needed. Decreasing this value helps in reducing memory requirements.

am.mp.methodStackSize

The default is 100. This value is the maximum stack size of any running thread that is recorded in method profiling.

am.mp.threadSize

The default is 1000. This value is the maximum running thread size that can be monitored at any instance of time.

dc.turbomode.enabled

The default setting is `true`, which enables turbo mode.

By default, the Data Collector limits the amount of native memory it uses to 100 MB, see the description of internal.memory.limit on page "internal.memory.limit" on page 56. The Data Collector enters turbo mode when the Data Collector native memory use exceeds 75% of the native memory limit, by default 75 MB. (You can adjust this percentage with turbo.mem.ulimit to adjust the percentage. However, do not set turbo.mem.ulimit unless directed by IBM Software Support.) The behavior when the memory utilization is below 75 MB is the same whether turbo mode is enabled or disabled.

Behavior when dc.turbomode.enabled is enabled and the Data Collector is in turbo mode

When the Data Collector switches to turbo mode, a message `Switching to Turbo Mode` is logged in the trace-dc-native.log file.

In turbo mode, the Data Collector stops monitoring new requests and holds existing requests. It also switches Network Agent and Event Agent threads to the higher priorities specified by the na.turbo.priority and ea.turbo.priority properties respectively. It also lowers the sleep time of the

Event Agent and Network Agent threads specified by the ea.turbo.sleep and na.turbo.sleep properties respectively. All these actions are done to drain the native memory quickly by sending accumulated event data to the Publish Server.

In turbo mode, if a new request comes in, the Data Collector simply does not monitor the new request. It continues to monitor the already running requests. The Data Collector notifies the Publish Server that a new request was not monitored when in turbo mode. A notification is sent to the Managing Server for every new request that is not monitored by sending a dropped record. The Publish Server in turn reflects this status in Publish Server corrupted request counters obtained through `amctl.sh ps1 status`.

When turbo mode is enabled, data in the Application Monitor user interface is always accurate. The accuracy comes at the cost of pausing application threads for a few seconds.

Behavior when dc.turbomode.enabled is enabled and the Data Collector is in normal mode

The Data Collector switches back to normal mode, when the Data Collector native memory use falls below 75% of the limit. When the switch to normal mode happens, the Data Collector releases the requests that were placed on hold while switching to turbo mode. The Data Collector resumes monitoring all requests from then on.

When the Data Collector switches to normal mode, a message `Switching to Normal Mode` is logged in the trace-dc-native.log file. It also logs memory utilization and a time stamp.

Behavior when dc.turbomode.enabled is disabled

A value of `false` disables turbo mode. When turbo mode is disabled, the Data Collector does not pause the application thread when the native memory use exceeds 75% of the limit. Instead, it drops the accumulated diagnostic data instead of sending it to the Managing Server. Therefore, the data shown in the Application Monitor user interface is incomplete. But the response time of the application threads is not negatively impacted. An appropriate message indicating data is dropped is logged in msg-dc-native.log and trace-dc-native.log. The Managing Server discards all the diagnostic data gathered for the request when the Data Collector drops records related to that request.

Disabling dc.turbomode.enabled

The default setting is `true`, which enables turbo mode.

If any of the following conditions apply, disable turbo mode by setting dc.turbomode.enabled to `false`:

- Within the first 10 minutes after starting the Data Collector, it goes into turbo mode (search for the message `Switching to Turbo Mode` in trace-dc-native.log).
- You do not want your applications to be paused temporarily as the Data Collector native memory exceeds 75% of the limit. Disabling turbo mode comes at the cost of losing the monitoring data when this boundary condition is reached.

An alternative is increasing the internal.memory.limit to allow more native memory use. This increase is done at the risk of requesting more native

memory from the JVM than what is available. In this event, the JVM issues OutOfMemory errors. See the description of internal.memory.limit on page "internal.memory.limit" on page 56.

The toolkit properties file

The toolkit properties file is automatically created by the data collector at startup, using various input files. It is unique for every application server instance monitored by the data collector. It is located in the instance directory and its name is Its name is `toolkit.properties`.

Because this file is re-created at each data collector startup, **do not make any changes** to this file; if you do, they will be overwritten.

Instead, add the settings that you want to modify to the toolkit custom properties file. This file is located in the `custom` subdirectory of the instance directory. Its name is `toolkit_custom.properties`.

Important: If the `toolkit_custom.properties` file does not exist, create it when you want to make changes. You might also have to create the `custom` subdirectory in the instance directory.

Configuring the Data Collector after changing the application server version

If you change the version of the application server being monitored by the Data Collector, you must reconfigure the Data Collector to point to the updated instance of the application server.

Complete the following steps:

1. Log on to the computer where you installed the Data Collector using the user that performed the installation.
2. Start the instance of the application server that is being monitored by the Data Collector.
3. Use the configuration tool to unconfigure the Data Collector. See "Unconfigure the Data Collector from application server instances" on page 41 for instructions.
4. Use the configuration tool to configure the Data Collector again. See "Configure the Data Collector to monitor application server instances" on page 36 for instructions.
5. Restart the instance of the application server that is being monitored by the Data Collector.

Changing the IP address of the Data Collector host computer

To change the IP address of the Data Collector host computer, perform the following procedure:

1. Use the configuration tool to unconfigure the Data Collector. See "Unconfigure the Data Collector from application server instances" on page 41 for instructions.
2. If the instance of the application server that is being monitored by the Data Collector is not stopped, stop it.
3. Perform the IP address change at the operating system and network level.

4. Use the configuration tool to configure the Data Collector again. See "Configure the Data Collector to monitor application server instances" on page 36 for instructions.
5. If the instance of the application server that is being monitored by the Data Collector is not started, start it.

Moving the Data Collector to a different host computer

The following prerequisites are required if you want to move the Data Collector to a different host computer while keeping the same Probe ID and Controller ID:

- Host A and host B must have the same configuration at the operating system level.
- You must move the same version of the Data Collector from host A to host B.

To maintain the Probe ID and Controller ID when moving to another physical host, you need to use the ID file:

Table 8. Locations of the ID file

WebLogic	If the monitored server instance is represented by a weblogic machine: *DC_home*/runtime/wls*app_server_version.domain_name.machine_name.instance_name*/id else: *DC_home*/runtime/wls*app_server_version.domain_name.host_name.instance_name*/id
Tomcat	*DC_home*/runtime/tomcat*app_server_version.host_name.instance_name*/*DC_home*/runtime/id
JBoss	*DC_home*/runtime/jboss*app_server_version.host_name.instance_name*/id
NetWeaver	*DC_home*/runtime/netweaver*app_server_version.sap_node_ID_host_name.sap_instance_number*/id
J2SE	*DC_home*/runtime/j2se.*application_name.host_name.instance_name*/*DC_home*/runtime/id

Perform the following procedure:

1. On host A, stop the instance of the application server that is being monitored by the Data Collector.
2. On host B, install the Data Collector and configure it using the Application Monitor user interface. Configuring the Data Collector generates the ID file and other Data Collector runtime property files.
3. On host B, unconfigure the Data Collector. This step deletes all information about this Data Collector from the ITCAM for J2EE database.
4. On host B, stop the instance of the application server that is being monitored by the Data Collector.
5. Copy the contents in the ID file of host A to the ID file of host B.
6. On host B, save the ID file.
7. On host B, start the instance of the application server that is being monitored by the Data Collector.

The Data Collector on host B assumes the identity of the Data Collector on host A and is configured with the runtime configuration of the Data Collector on host A.

Controlling Instrumentation of Application Classes for Memory Leak, Lock, and L3 Method Analysis

ITCAM for J2EE uses a technique called Byte Code Instrumentation (BCI). BCI collects Level 3 tracing data, Memory Leak Diagnosis data, and Lock Contention data from your applications. BCI is enabled by adjusting properties in the *custom_directory*/toolkit_custom.properties file.

Making these adjustments activates the use of one or more configuration files in the *DC_home*/itcamdc/etc directory, which contain the default settings to control BCI. The configuration files are described in the following table:

Table 9. BCI Configuration Files

File Name	Purpose	Default Behavior
method_entry_exit.xml	Defines application method entry and exit BCI	All non-trivial methods for all application classes are Byte-Code-Instrumented for method entry and exit analysis.
memory_leak_diagnosis.xml	Defines application Memory Leak Diagnosis BCI	Heap allocations for all classes instantiated by all application classes are Byte-Code-Instrumented.
lock_analysis.xml	Defines application lock analysis BCI	Lock acquire and release requests for all application classes are Byte-Code-Instrumented.

If you want to enable one or more of the BCI features with the default settings, see "Enabling BCI features with default settings."

If you want to customize the default settings and choose what classes and methods to modify, see one or more of the following sections:

- "Customizing method profiling and method entry and exit tracing" on page 63
- "Customizing Memory Leak Diagnosis" on page 65
- "Customizing Lock Analysis" on page 67

Enabling BCI features with default settings

Perform the following procedure to enable one or more of the BCI features with the default settings:

1. In the *custom_directory*/toolkit_custom.properties file, uncomment one or more of the following lines by removing the number sign (#) at the beginning of the line:

   ```
   am.camtoolkit.gpe.customxml.L3=DC_home/itcamdc/etc/method_entry_exit.xml
   am.camtoolkit.gpe.customxml.leak=DC_home/itcamdc/etc/memory_leak_diagnosis.xml
   am.camtoolkit.gpe.customxml.lock=DC_home/itcamdc/etc/lock_analysis.xml
   ```

   See Table 9 for a description of the default behaviors when each of these configuration files is activated.

2. Set one or more of the following properties to true:

   ```
   com.ibm.tivoli.itcam.toolkit.ai.enablememoryleakdiagnosis=true
   com.ibm.tivoli.itcam.toolkit.ai.methodentryexittrace=true
   com.ibm.tivoli.itcam.toolkit.ai.enablelockanalysis=true
   ```

Customizing method profiling and method entry and exit tracing

Method profiling and method entry and exit tracing are enabled together and use the same call interceptions. Method profiling is performed at MOD L2, and method entry and exit tracing is performed at MOD L3. You can configure the data collector to change the thresholds and limits for method profiling, and to exclude some classes and methods for method entry and exit tracing.

Customizing thresholds for Level 2 method profiling

The data collector only instruments method profiling data when the method exceeds certain thresholds of processor time and real ("wall clock") time usage. The total number of methods, stack size, and running thread size are also limited. You can customize the thresholds and limits.

The following properties in the data collector properties file control the thresholds and limits for method profiling. For more information about the file, see "Fine-tuning the data collector properties files" on page 55.

am.mp.cpuThreshold
The default is `30` milliseconds. Only the methods that take at least the minimum amount of processor time specified in this property are captured for method profiling data. This avoids unnecessary clutter. Generally, methods with greater than the value that is specified in this property are considered useful. Customers can reduce or increase this value if required.

am.mp.clockThreshold
The default is `30` milliseconds. Only the methods that take at least the minimum amount of wall clock time specified in this property are captured for method profiling data. This avoids unnecessary clutter. Generally, methods with greater than the value that is specified in this property are considered useful. Customers can reduce or increase this value if required.

am.mp.leagueTableSize
The default is `1000`. This is the maximum number of methods that are monitored for method profiling data. Customers can reduce or increase this value if required. Decreasing it helps to reduce memory requirements.

am.mp.methodStackSize
The default is `100`. This is the maximum stack size of any running thread that is recorded in method profiling.

Setting classes and methods for Level 3 method entry and exit tracing

By default, method entry and exit tracing on MOD L3 is performed for all classes and methods. To set specific classes and methods for method entry and exit analysis, complete the following procedure:

1. Make a copy of the *DC_home*`/itcamdc/etc/method_entry_exit.xml` file in a temporary location. Open the copy in a text editor.
2. Modify the parameters in the file. The following table describes the parameters that you can modify:

Table 10. Parameters for the Level 3 method entry and exit analysis configuration file

Tag name	Description
methodSelection	Defines the classes and methods to be modified. By default, all classes and methods are selected. By modifying the className and methodName tags within the methodSelection tag, you can implement a more granular selection. Each methodSelection tag must contain exactly one className tag, and one or more methodName tags. Multiple methodSelection tags can be specified.
className	Identifies the name of a class or classes to be modified. Each methodSelection tag must contain exactly one className tag.
methodName	Identifies a method or method within the class or classes identified by the className tag to be modified for entry/exit tracing. Each methodSelection tag must contain one or more methodName tags.

Both className and methodName tags can include wildcard characters. The following section describes how the wildcard characters work:

- Asterisk (*) stands for zero or more occurrences of any character when used by itself. When it is embedded within a sequence of characters (for example, java.*.String), it matches zero or more occurrences of any character except the package separator (.).
- Two periods (..) can be used to specify all subpackages. It matches any sequence of characters that starts and ends with the package separator (.).

  For example, java..String matches java.lang.String and com.ibm..* matches any declaration beginning with com.ibm.
- If the method name begins with an exclamation point (!), any methods that match the method name are excluded from BCI for entry and exit tracing. This is useful for indicating that all methods within a class or group of classes are to be Byte-Code-Instrumented except for those methods that are excluded.

For example, an application with a package name of com.mycompany.myapp has the following requirements:

- Within the Customer class, all methods must be Byte-Code-Instrumented.
- Within the Supplier class, all methods must be Byte-Code-Instrumented except for those methods beginning with the get or set.

The following example shows the contents of the customized method_entry_exit.xml file that accomplishes this:

```
<aspect>
  <type>application</type>
  <name>com.ibm.tivoli.itcam.toolkit.ai.aspectj.apptrace.EntryExitAspect</name>
  <enabledProperty>
         com.ibm.tivoli.itcam.toolkit.ai.methodentryexittrace</enabledProperty>
   <defaultEnabled>true</defaultEnabled>
   <methodSelection>
        <className>com.mycompany.myapp.Customer</className>
        <methodName>*</methodName>
   </methodSelection>
   <methodSelection>
        <className>com.mycompany.myapp.Supplier</className>
        <methodName>!get*</methodName>
        <methodName>!set*</methodName>
   </methodSelection>
</aspect>
```

3. Complete one of the following steps:

- Save the file in the *DC_home*/runtime/*app_server_version.node_name.server_name*/custom directory. Complete the following steps:
  a. In the toolkit_custom.properties file in the *DC_home*\runtime*appname.instname.hostname.dcname*\custom directory, set the property am.camtoolkit.gpe.customxml.L3 to the name (without path) of the file that you modified in Step 2 on page 63.
  b. In the same toolkit custom properties file, set the following property to true:

     ```
     com.ibm.tivoli.itcam.toolkit.ai.methodentryexittrace=true
     ```
- Save the file in any directory on the monitored host. Complete the following steps:
  a. In the toolkit_custom.properties file, set the property am.camtoolkit.gpe.customxml.L3 to the path and name for the file that you modified in Step 2 on page 63. .
  b. Set the following property to true:

     ```
     com.ibm.tivoli.itcam.toolkit.ai.methodentryexittrace=true
     ```

Customizing Memory Leak Diagnosis

Perform the following procedure to enable Memory Leak Diagnosis with customized settings:

1. Make a copy of the *DC_home*/itcamdc/etc/memory_leak_diagnosis.xml file, and open it up in a text editor.
2. Modify the parameters in the memory_leak_diagnosis.xml file. The following is a description of the parameters you can modify:

Table 11. Parameters for Memory Leak Diagnosis Configuration File

Tag Name	Description
heapAllocationTarget	Defines the allocating and allocated classes for which heap allocations will be Byte-Code-Instrumented. By default, all allocating and allocated classes are selected. By modifying the allocatingClassName and allocatedClassName tags within the heapAllocationTarget tag, you can implement a more granular selection. Each heapAllocationTarget tag must contain exactly one allocatingClassName tag, and one or more allocatedClassName tags. Multiple heapAllocationTarget tags can be specified.
allocatingClassName	Identifies the name of a class or classes to be modified. Each heapAllocationTarget tag must contain exactly one allocatingClassName tag.
allocatedClassName	Identifies the specific heap allocation requests within the class or classes identified by the allocatingClassName tag that are to be Byte-Code-Instrumented. Each heapAllocationTarget tag must contain one or more allocatedClassName tags.

Both the allocatingClassName and the allocatedClassName tags can include wildcard characters. The following summary describes how the wildcard characters work:

- Asterisk (*) stands for zero or more occurrences of any character when used by itself. When embedded within a sequence of characters (for example, java.*.String), it matches zero or more occurrences of any character except the package separator (.).
- Two periods (..) can be used to specify all sub-packages (for example, java..String matches java.lang.String). It matches any sequence of characters that starts and ends with the package separator (.).
- If the allocated class name begins with an exclamation point (!), any heap allocations for classes that match the allocated class name are specifically

excluded from BCI for Memory Leak Diagnosis. This is useful for indicating that all heap allocations within a class or group of classes are to be Byte-Code-Instrumented except for those allocations that are specifically excluded.

For example, an application with a package name of com.mycompany.myapp has the following requirements:

- Within the Customer class, all heap allocations should be Byte-Code-Instrumented.
- Within the Supplier class, all heap allocations should be Byte-Code-Instrumented except for allocations for classes beginning with `java.lang.String`.

The following example describes the contents of the customized memory_leak_diagnosis.xml file that accomplishes this:

```
<aspect>
    <type>application</type>
    <name>com.ibm.tivoli.itcam.toolkit.ai.aspectj.apptrace.CaptureHeap</name>
    <enabledProperty>
                com.ibm.tivoli.itcam.toolkit.ai.enablememoryleakdiagnosis</enabledProperty>
    <defaultEnabled>true</defaultEnabled>
    <-- Modify the heapAllocationTarget tag to select or deselect the allocating and
          allocated classes for Memory Leak Diagnosis -->
    <heapAllocationTarget>
        <allocatingClassName>
                    com.mycompany.myapp.Customer</allocatingClassName>
        <allocatedClassName>*</allocatedClassName>
    </heapAllocationTarget>
    <heapAllocationTarget>
        <allocatingClassName>
                    com.mycompany.myapp.Supplier</allocatingClassName>
        <allocatedClassName>!java.lang.String*</allocatedClassName>
    </heapAllocationTarget>
</aspect>
```

3. Complete one of the following steps:
   - Save the file in *custom_directory*, then complete the following steps:
     a. In the *custom_directory*/toolkit_custom.properties file, uncomment the following line by removing the number sign (#) at the beginning of the line:

        ```
        am.camtoolkit.gpe.customxml.leak=DC_home/itcamdc/etc/
        memory_leak_diagnosis.xml
        ```

     b. Change this line by replacing the path with just the file name of the file you modified in Step 2 on page 65.
     c. Set the following property to true:

        ```
        com.ibm.tivoli.itcam.toolkit.ai.enablememoryleakdiagnosis=true
        ```

   - Save the file in any directory on your server, then complete the following steps:
     a. In the *custom_directory*/toolkit_custom.properties file, uncomment the following line by removing the number sign (#) at the beginning of the line:

        ```
        am.camtoolkit.gpe.customxml.leak=DC_home/itcamdc/etc/
        memory_leak_diagnosis.xml
        ```

     b. Change this line by specifying the path and name for the file you modified in Step 2 on page 65.
     c. Set the following property to true:

        ```
        com.ibm.tivoli.itcam.toolkit.ai.enablememoryleakdiagnosis=true
        ```

Customizing Lock Analysis

Perform the following procedure enable lock analysis with customized settings:

1. Make a copy of the *DC_home*/itcamdc/etc/lock_analysis.xml file, and open it up in a text editor.
2. Modify the lockingClasses parameter in the lock_analysis.xml file.

   The parameter defines the classes for which lock requests will be Byte-Code-Instrumented. By default, all lock requests in all application classes are selected. By modifying this tag, you can implement a more granular selection, although within a class all lock requests are Byte-Code-Instrumented. Multiple lockingClasses tags can be specified.

   The lockingClasses tag can include wildcard characters. The following summary describes how the wildcard characters work:

   - Asterisk (*) stands for zero or more occurrences of any character when used by itself. When embedded within a sequence of characters (for example, `java.*.String`), it matches zero or more occurrences of any character except the package separator (.).
   - Two periods (..) can be used to specify all sub-packages (for example, `java..String` matches `java.lang.String`). It matches any sequence of characters that starts and ends with the package separator (.).
   - If the locking class name begins with an exclamation point (!), any classes matching the classes identified in the tag are specifically excluded from BCI for lock analysis. This is useful for indicating that all classes are to be Byte-Code-Instrumented except for those classes that are specifically excluded.

   For example, an application with a package name of com.mycompany.myapp has the following requirements:

   - Only classes that begin with `Cus` or `Sup` should be Byte-Code-Instrumented for lock analysis.
   - The Supplier class should not be Byte-Code-Instrumented for lock analysis.

   The following would be the contents of the customized lock_analysis.xml file that accomplishes this:

```
 <aspect>
   <type>application</type>
   <name>com.ibm.tivoli.itcam.toolkit.ai.aspectj.apptrace.CaptureLock</name>
   <enabledProperty>
           com.ibm.tivoli.itcam.toolkit.ai.enablelockanalysis</enabledProperty>
   <defaultEnabled>true</defaultEnabled>
   <lockingClass>com.mycompany.myapp.Cus*</lockingClass>
   <lockingClass>com.mycompany.myapp.Sup*</lockingClass>
   <lockingClass>!com.mycompany.myapp.Supplier</lockingClass>
</aspect>
```

3. Complete one of the following steps:
   - Save the file in *custom_directory*/, then complete the following steps:
     a. In the *custom_directory*/toolkit_custom.properties file, uncomment the following line by removing the number sign (#) at the beginning of the line:

        `am.camtoolkit.gpe.customxml.lock=`*DC_home*`/itcamdc/etc/lock_analysis.xml`

     b. Change this line by replacing the path with just the file name of the file you modified in Step 2.
     c. Set the following property to true:

        `com.ibm.tivoli.itcam.toolkit.ai.enablelockanalysis=true`

- Save the file in any directory on your server, then complete the following steps:
  a. In the *custom_directory*/toolkit_custom.properties file, uncomment the following line by removing the number sign (#) at the beginning of the line:

     ```
     am.camtoolkit.gpe.customxml.lock=DC_home/itcamdc/etc/lock_analysis.xml
     ```

  b. Change this line by specifying the path and name for the file you modified in Step 2 on page 67.
  c. Set the following property to true:

     ```
     com.ibm.tivoli.itcam.toolkit.ai.enablelockanalysis=true
     ```

Note: See the Monitoring on Demand chapter of the *IBM Tivoli Composite Application Manager: User's Guide* for a description of monitoring levels and information about how to manage monitoring levels.

Setting the Heap Dump scan interval and logging

The Heap Dump Management function of ITCAM for J2EE can create Heap Dumps of the monitored IBM WebSphere Application Server by user request.

Once in a defined time interval, ITCAM for J2EE scans the existing Heap Dumps, to inform the user of their existence. This scan also serves to delete heap dump files that are over 48 hours old.

By default, this interval is every 12 hours. To change the interval, set the following property in the *custom_directory*/toolkit_custom.properties file to the new interval in seconds:

```
am.mddmgr.poll.delay
```

To enable logging of heap Dump scans, set the following property in the cynlogging.properties file. This file is located in the directory that also contains *custom_directory*:

```
CYN.trc.datacollector.level=DEBUG_MIN
```

Once every scan interval (12 hours by default), Heap Dump scan messages are logged in to the trace-dc-ParentLast.log file.

Defining custom requests

A custom request is an application class and method that you designate as an edge or nested request. When the method runs, a start and end request trace record is written to the Level 1 or Level 2 tracing.

Custom requests are defined in the *DC_home*/itcamdc/etc/custom_requests.xml file. The product-supplied version of this file is only a sample and must be customized by the user. In addition, this feature is enabled by adjusting properties in the *custom_directory*/toolkit_custom.properties file.

Perform the following procedure to enable and define tracing of custom requests:

1. Make a copy of the custom_requests.xml file, and open it up in a text editor.
2. Modify the parameters in the custom_requests.xml file. The following table describes the parameters you can modify:

Table 12. Parameters for Custom Requests Configuration File

Tag Name	Description
edgeRequest	Identifies one or more application methods that are to be Byte-Code-Instrumented for custom request processing. By modifying the requestName, Matches, type, and methodName tags within the edgeRequest tag, you can customize the selection. Each edgeRequest tag must contain exactly one methodName tag, and one or more Matches tags. Multiple edgeRequest tags can be specified.
requestName	Defines a unique name for this request. The request name is displayed in the L1 or L2 trace entry that is produced when one of the methods identified by this custom request runs.
Matches	Identifies a class or classes that contain the methods that are to be Byte-Code-Instrumented for custom request processing. Multiple Matches tags can be present within a single edgeRequest tag.
type	Indicates whether a class must be a system or application class to match the edgeRequest tag.
methodName	Identifies the names of the methods within one of the classes identified by the Matches tag that are to be Byte-Code-Instrumented for custom request processing. Exactly one methodName tag can be specified in each edgeRequest tag.

The Matches and the methodName tags can include wildcard characters. The following section describes how the wildcard characters work:

- Asterisk (*) stands for zero or more occurrences of any character when used by itself. When embedded within a sequence of characters (for example, `java.*.String`), it matches zero or more occurrences of any character except the package separator (.).
- Two periods (..) can be used to specify all subpackages (for example, `java..String` matches `java.lang.String`). It matches any sequence of characters that starts and ends with the package separator (.).

For example, an application with a package name of com.mycompany.myapp has the following requirements:

- Within the Customer class, treat the creditCheck() method as a custom request called CreditCheck.
- Within the Supplier class, treat the inventoryCheck() method as a custom request called SupplyCheck.

The following example shows the contents of the customized custom_requests.xml file that accomplishes these requirements:

```
<customEdgeRequests>
      <edgeRequest>
              <requestName>CreditCheck</requestName>
              <Matches>com.mycompany.myapp.Customer</Matches>
              <type>application</type>
              <methodName>creditCheck</methodName>
      </edgeRequest>
      <edgeRequest>
              <requestName>SupplyCheck</requestName>
              <Matches>com.mycompany.myapp.Supplier</Matches>
              <type>application</type>
              <methodName>inventoryCheck</methodName>
      </edgeRequest>
</customEdgeRequests>
```

3. Complete one of the following steps:
   - Save the file as *DC_home*/itcamdc/etc/custom_requests.xml, then complete the following steps:
     a. In the *custom_directory*/toolkit_custom.properties file, uncomment the following line by removing the number sign (#) at the beginning of the line:

```
#am.camtoolkit.gpe.customxml.custom=DC_home/itcamdc/etc/
custom_requests.xml
```

b. Change this line by replacing the path with just the file name of the file you modified in Step 2 on page 68.

- Save the file in any directory on your server, then complete the following steps:
  a. In the *custom_directory*/toolkit_custom.properties file, uncomment the following line by removing the number sign (#) at the beginning of the line:

     ```
     #am.camtoolkit.gpe.customxml.custom=DC_home/itcamdc/etc/
     custom_requests.xml
     ```

  b. Change this line by specifying the path and name for the file you modified in Step 2 on page 68.

Disabling various types of Byte Code Instrumentation for J2EE APIs

The Data Collector uses a technique called Byte Code Instrumentation (BCI) to collect data from various types of J2EE APIs that typically operate as nested requests. BCI is automatically enabled for these types of APIs. It can be disabled by adding lines to the *custom_directory*/toolkit_custom.properties file.

Disable instrumentation of one or more of the following types of APIs by adding the following lines to the toolkit_custom.properties file:

Table 13. Adding lines to toolkit_custom.properties

Type of J2EE API	Line to add to toolkit_custom.properties file
Enterprise JavaBeans (EJB)	`com.ibm.tivoli.itcam.toolkit.ai.enableejb=false`
Java Connector Architecture (JCA)	`com.ibm.tivoli.itcam.toolkit.ai.enablejca=false`
Java Database Connectivity (JDBC)	`com.ibm.tivoli.itcam.toolkit.ai.enablejdbc=false`
Java Naming and Directory Interface (JNDI)	`com.ibm.tivoli.itcam.toolkit.ai.enablejndi=false`
Java Message Service (JMS)	`com.ibm.tivoli.itcam.toolkit.ai.enablejms=false`
Servlets/ JavaServer Pages (JSP)	`com.ibm.tivoli.itcam.toolkit.ai.enableservlet=false`
HTTP session count tracking	`com.ibm.tivoli.itcam.toolkit.ai.enablesessioncount=false`
CICS Transaction Gateway (CTG)	`com.ibm.tivoli.itcam.dc.ctg.enablectg=false`
IMS	`com.ibm.tivoli.itcam.dc.mqi.enableims=false`
Java Data Objects (JDO)	`com.ibm.tivoli.itcam.dc.mqi.enablejdo=false`
Message Queue Interface (MQI)	`com.ibm.tivoli.itcam.dc.mqi.enablemqi=false`

Table 13. Adding lines to toolkit_custom.properties (continued)

Type of J2EE API	Line to add to toolkit_custom.properties file
Axis web service (only on JBoss and WebLogic)	`com.ibm.tivoli.itcam.toolkit.ai.axis.enablewebservice=false`
Remote Method Invocation (RMI)	`am.ejb.rmilistener.enable=false`

For performance reasons, you can also disable BCI for several API types only for Level 1 monitoring. In this case, BCI will for the API types be enabled only when the monitoring level is set to 2 or 3.

To do this, add (or uncomment) the following lines in the *custom_directory*/toolkit_custom.properties file.

Table 14. Modifying lines in toolkit_custom.properties

Type of J2EE API	Line to add to toolkit_custom.properties file
JCA	`com.ibm.tivoli.itcam.toolkit.ai.jca.callback.unconditional=false`
JDBC	`com.ibm.tivoli.itcam.toolkit.ai.jdbc.callback.unconditional=false`
JNDI	`com.ibm.tivoli.itcam.toolkit.ai.jndi.callback.unconditional=false`
JMS	`com.ibm.tivoli.itcam.toolkit.ai.jms.callback.unconditional=false`

Specifying data collection for custom MBeans

If you have custom MBeans, customize the generic configuration for Java Management Extensions (JMX) data collection.

Perform the following procedure to customize the generic configuration for JMX data collection:

1. The following table describes the parameters you can use:

Table 15. Parameters for JMX MBean Configuration file

Element Name	Sub-element Name	Description
DomainList	Version	Defines the version of the application server
Domain	Name	Defines a domain. If the asterisk (*) is defined, all MBeans that match the query "ObjectName" will be returned. Otherwise, the MBeans that belong only to this domain name will be returned.
Domain	Description	Describes the domain. This can be any text string.
Domain	MBean	Defines the MBeans to be collected
MBean	ObjectName	Defines the MBean object name for collection. If the MBean element is used within an Attr element (which indicates the embedded MBean), then the object name is any symbolic name, such as $ATTRIBUTE_VALUE. This symbolic name will be replaced with the actual object name internally.
MBean	Category	Defines a unique key for the MBean. Each MBean must have a unique key, which is used in the JMXAcquireAttribute to get the MBean attributes.
MBean	RetrieveAllAttrs	A value of true indicates that all the attributes for the MBean must be collected. There is no need to define the attributes in the Attr element.

Table 15. Parameters for JMX MBean Configuration file (continued)

Element Name	Sub-element Name	Description
MBean	Attr	Defines the attributes to be collected
Attr	Name	The attribute name
Attr	MappedKey	Defines a unique key for the attribute. Each attribute must have a unique key, which is used in the JMXAcquireAttribute to get the specific attribute.
Attr	MBean	Defines the embedded MBean within this attribute. This tag is used when an attribute has an embedded MBean, which points to another MBean with the object name.
Attr	JavaBean	Defines the embedded MBean within this attribute. This tag is used when an attribute has an embedded MBean, which points to another MBean object. The object is the java object, which has the elements of a JavaBean (setter, getter).
Attr	TargetType	Defines the type of the attribute. This is usually specified for the JavaBean type to determine the attribute type.

2. The following example shows a customized MBean configuration file:

```
<DomainList>
  <Version>1.0.0</Version>
  <Domain>
    <Name>*</Name>
    <Description>Custom MBean Conf</Description>
    <JSR77Compliant>false</JSR77Compliant>
     <MBean>
      <ObjectName>type=OperatingSystem,*</ObjectName>
      <Category>OPERATINGSYSTEM</Category>
      <Attr>
        <Name>Arch</Name>
        <MappedKey>OPERATE_ARCH</MappedKey>
      <Attr>
      <Attr>
        <Name>AvailableProcessors</Name>
        <MappedKey>OPERATE_AVAILABLE</MappedKey>
      <Attr>
     </MBean>
    <MBean>
     <ObjectName>type=Runtime,*</ObjectName>
     <Category>JVM</Category>
     <Attr>
       <Name>StartTime</Name>
       <MappedKey>JVM_STARTTIME</MappedKey>
     <Attr>
    </MBean>
  </Domain>
</DomainList>
```

3. Complete one of the following steps:
   - Save the file as *DC_home*/itcamdc/etc/custom_mbeanconfig.xml, and complete the following steps:
     a. Open the *custom_directory*/toolkit_custom.properties file.
     b. Uncomment the line beginning with am.camtoolkit.jmxe.custom by removing the number (#) sign.
   - Save the file in any directory on your server, then complete the following steps:
     a. Open the *custom_directory*/toolkit_custom.properties file.
     b. Uncomment the line beginning with am.camtoolkit.jmxe.custom by removing the number (#) sign.

c. Change this line by specifying the path and name for the file you modified in Step 1 on page 71.

Specifying data collection for custom MBeans - an alternative approach

The *custom_directory*/toolkit_custom.properties file contains the following properties with their default values:

```
am.getallmbeans=y
am.jmxkeyword=type_identifier
am.jmxusecanonical=y
am.jmxtruncate=n
am.jmxlength=30
```

These properties are in effect, only if the custom MBeans property is commented out in the toolkit_custom.properties file, as shown in the following example:

```
#  Uncomment the line below to enable custom mbeans
#am.camtoolkit.jmxe.custom=[file_path]/custom_mbeanconfig.xml
```

The presence of these properties displays all the existing MBeans in the application server, except for the ones that are already part of the mbeanconfig xml file. This is the list of the properties and their definitions:

am.getallmbeans

You can use this property to get all the existing MBeans in the application server except for those that are already defined in the mbeanconfig.xml file. This property is in effect while the custom MBeans property is not set. If the custom MBeans property is set, the property has no effect on getting all the MBeans. Set its value to "y" to activate it.

By default, the keyword "type" or "Type" is searched within each acquired object name. Having the domain name and the value of the "type/Type" creates the category name. The category name is displayed on the System Resource page on the Visualization Engine. If "type" or "Type" does not exist, the "name" keyword is searched in the object name, and its value is used to create the category name. If the "name" keyword does not exist, the canonical name that contains all the keywords for the object name is used.

am.jmxkeyword

If for some reason the 'type' or 'Type' keyword does not distinguish the MBeans, and you need more granularity, then you have to define more keywords to be included in the category name.

For example, if you specify the keyword "identifier" in addition to the "type/Type" keyword the value of the "identifier" will be included in the category name. The category name includes the "type/Type" value and the "identifier" value separated by an underscore (_) character. More than one keyword can be specified in the property. The keywords must be separated by a comma (,).

am.jmxusecanonical

If for some reason, you need to see the entire keywords in the object name (this could be a long string, so you should avoid doing it), then assign the

"y" value to this property. This will result in including the entire keywords values for the category name separated by an underscore (_) character.

am.jmxtruncate

In some cases, especially in the case of using the canonical keyword, if the length of the category name is too long JMXEngine will automatically truncate its length to 30 characters. This is the default setting. If there is no need to truncate the category name, assign the "n" value to this property to prevent the truncation.

am.jmxlength

The default truncation length is 30 characters. If you want to have a different truncation length set it in this property. Values above and below "30" are accepted.

Customizing CICS transaction correlation

CICS is a transaction framework, primarily used to run mature applications. To communicate with CICS, Java applications can use the CICS Transaction Gateway (CTG).

ITCAM for J2EE can use BCI (Byte Code Instrumentation) to collect data on CTG calls. The BCI engine injects callback code into CTG classes. To enable this feature, set the following property in the *custom_directory*/toolkit_custom.properties file:

```
com.ibm.tivoli.itcam.dc.ctg.enablectg=true
```

By default, when CTG BCI is enabled, the Data Collector callback code adds composite tracking data, called Global Publish Server (GPS) tokens. This data is added into the communications area (COMMAREA) used to carry transaction request data to CICS. This data can be used by ITCAM for Transactions, which instruments the CICS transaction framework. ITCAM for Transactions correlates every CICS transaction with the corresponding CTG call using the GPS token. The user can then view a detailed breakdown of transaction response time in the ITCAM Visualization Engine.

However, the presence of the GPS token in COMMAREA might not always be desirable. Disable GPS tokens if ITCAM for Transactions is not used for the CICS server. Otherwise, the GPS token reaches the server application, which might (in some cases) not process it correctly.

You can selectively disable GPS tokens for specific transactions. Selections can be based on CTG gateway address or protocol; by CICS system; by CICS program, or by the CICS transaction ID. To selectively disable GPS tokens, edit the file *custom_directory*/ctg.filters . This file can contain any number of lines with the following syntax:

```
Type=E|I[,Gateway=<CTG URL>][,Server=<CICS Server>][,Program=<CICS Program>]
  [,Transid=<Mirror tran ID>]
```

Each line defines a filter, which disables or enables GPS tokens for some transactions.

The `Type` parameter is mandatory for each line. A value of "E" sets up an Exclude filter; transactions matching it do not have a GPS token inserted into the COMMAREA. "I" denotes an Include filter; any transactions matching an include filter have a GPS token, overriding any Exclude filter applying to them.

All other parameters are optional, but at least one of them must be present on every line. To match a filter, a transaction must match all of the parameters set on the line:

- `Gateway` is any part of the CTG URL, including the protocol, host name and/or port
- `Server` is the host name of the CICS server (this name might be different from the CTG host name)
- `Program` is the CICS program name (a field in a CICS transaction request)
- `Transid` is the CICS Mirror Transaction ID. Except Multi Regional Operation (MRO) CICS/CTG environments, this parameter is of little use as all CTG transactions have the same Mirror Transaction ID

For example, to disable addition of GPS tokens to the COMMAREA of all transactions routed through the local protocol, add the following line to *custom_directory*/ctg.filters:

```
Type=E,Gateway=local://*
```

To disable addition of GPS tokens to some transactions while enabling them for other transactions, use lines similar to the following example:

```
Type=E,Program=CYN$*,Server=CICS3101
/*Disables addition of GPS tokens to transactions for programs starting 'CYN$' to be
  run on the CICS3101 server.*/
Type=I,Program=CYN$ECI2,Server=CICS3101
/*Enables addition of GPS tokens for transactions for the CYN$ECI2 program to be
  run on the CICS3101 server.*/
```

To disable addition of GPS tokens to all transactions, use the following line:

```
Type=E,Gateway=*
```

Enabling instrumentation of Web Services as new request types

On the JBoss and Weblogic application servers, Web Services can be instrumented by the Data Collector. By default, this feature is disabled.

To enable instrumentation of Web Services as new request types, set (uncomment) the following property in the *DC_HOME*/runtime/*instance_name*/dc.properties file:

```
ws.instrument=true
```

Only JAX-RPC 1.1 and Axis 1.x Web services will be instrumented.

To enable Web Services correlation in the Visualization Engine and in ITCAM for Transactions, you need to instrument both the Web services client and the Web services server using ITCAM for WebSphere Data Collectors, and these Data Collectors must be connected to the same Managing Server.

Installing Memory Dump Diagnostic for Java with IBM Support Assistant

Memory Dump Diagnostic for Java (MDD for Java) either analyzes a single heap dump or analyzes and compares two heap dumps and searches for evidence of a memory leak. In order to download MDD for Java, you will need to first install IBM Support Assistant (ISA). ISA provides extra help with diagnosing problems and provides extra tools and components for troubleshooting as well as providing a place to write problems (PMRs).

MDD for Java analyzes manual or scheduled heap dumps performed by ITCAM's Heap Dump Management feature.

You can use ITCAM's Heap Dump Management feature to schedule or immediately initiate the collection of an IBM Heap Dump for a particular application server. Then this dump must be downloaded and post-processed outside ITCAM's user interface (Application Monitor) using MDD for Java. (The other Memory Diagnosis tools provided by ITCAM, such as Memory Analysis, Heap Analysis and Memory Leak Diagnosis, provide analysis via the Application Monitor.)

MDD for Java only analyzes heap dumps from IBM JDKs. For non-IBM JDKs use the ITCAM Memory Leak Diagnosis feature.

Searching capabilities are not supported for ITCAM for J2EE in ISA.

Where to Install ISA and MDD for Java

The following section describes two common configurations:

- Install ISA & MDD for Java on a standalone server that is not running an application server. After the IBM heap dump has been collected on the application server, it must be transferred to the MDD for Java server for post-processing.

  This configuration is recommended for production environments where you do not want the post-processing of the dump to impact the performance of the application server.
- Install ISA and MDD for Java on each application server host computer, so that you can analyze the heap dump locally without having to transfer it.

  This configuration may be suitable for a development or test environment where the overhead of analyzing the heap dump is not a concern.

The decision on where to install may also be influenced by the platforms supported by ISA.

Downloading, installing, configuring, and launching ISA

See the online helps in the Managing Server's user interface (Application Monitor) for instructions on how to download, install, configure, and launch ISA and to install the ISA plugin. Go to **Help > Welcome > Using IBM Support Assistant to diagnose problems**.

Note: ISA can be installed on both the Data Collector and Managing Server servers, but only the ISA installed on the Managing Server server can be invoked from the user interface (Application Monitor).

Installing MDD for Java

See the online helps in the Managing Server's user interface (Application Monitor) for instructions on how to install MDD for Java. Go to **Help > Welcome > > Memory Diagnosis > Heap Dump Management > Downloading Memory Dump Diagnostic for Java from IBM Support Assistant**.

Note: To download MDD for Java from ISA, the server where ISA is running needs to access the IBM Web site.

Configuring a Data Collector for multiple network cards and NATs

If a Data Collector needs to expose a specific IP to the Managing Server, complete one of the following steps:

- If the Data Collector is not using Port Consolidator, complete the following steps:
  1. Specify a system property java.rmi.server.hostname for the application server and set it to the IP address of the Data Collector.
  2. Make sure that Managing server can access the IP address of the Data Collector (You can verify this by doing a ping).
- If the Data Collector is using Port Consolidator, complete the following steps:
  1. Specify a system property java.rmi.server.hostname for the application server and set it to the IP address of the Data Collector.
  2. Specify a system property java.rmi.server.hostname in the `start` section of the script used to start Port Consolidator and set its value to the IP address of the Data Collector.
  3. Make sure that the Managing server can access the IP address of the Data Collector (You can verify this by doing a ping).

Parameters specified with multiple network cards

To install multiple network cards: make sure that the IP specified for the Data Collector server are IPs that can be used to communicate with each other (In other words, if there is a network configuration where one of the IPs does not have a path to the other server, do not use that IP).

Complete the following steps:

1. On the Data Collector servers, define an additional Java system property and set it to the IP address of the Data Collector:

   ```
   java.rmi.server.hostname
   ```

2. On the Data Collector server, in the *custom_directory*/datacollector_custom.properties file set kernel.codebase and kernel.rfs.address parameters to point to the Managing Server IP.
3. On the Data Collector host, in the *instance_runtime_directory* open any existing generated Data Collector property files (named `*datacollector.properties`). Delete kernel.codebase and kernel.rfs.address parameters from these files, if they are present.
4. Start the instance of the application server that will be monitored by the Data Collector.

Suppressing verbose garbage collection output in Data Collectors with a Sun JDK

For Sun JDKs, the Data Collector configuration enables verbose garbage collection output using the -Xloggc generic JVM argument. By default, the -Xloggc causes the JVM to generate class loading and unloading events to the native standard output stream. The process might fill the log files and consume excessive disk space.

To suppress class loading and unloading events, use your application server to add the -XX:-TraceClassUnloading -XX:-TraceClassLoading options to the arguments for the Java Virtual Machine. Then, Restart the instance of the application server that is being monitored by the Data Collector.

Configuring the Tomcat Data Collector to run as a Windows service

Once you have configured the Data Collector, you can complete the following steps to configure the Tomcat Data Collector to run as a Windows service.

1. Open the <AppServer_home>/bin/catalina.bat file.
2. Right-click the Tomcat Service icon on the Windows taskbar and click Configure.
3. When the Apache Tomcat properties window opens, click the Java tab.
4. From the open catalina.bat file, copy the value for JAVA_OPTS, and paste it into the text box labeled Java Options (in the Apache Tomcat Properties window).
5. Then paste the following text into the text box labeled Java Options, copying the exact settings from the JAVA_OPTS line of the catalina.bat file::

   ```
   Xbootclasspath/p:
   %PRODUCT_HOME%\itcamdc
   \lib\ext\tomcat\bcm\tomcat.bcm.jar -Dam.appserver=%APPSERVER% -Dam.nodename
   =%NODENAME% -Djava.rmi.server.RMIClassLoaderSpi=com.ibm.tivoli.itcam.tomcat
   .sdc.DCRMIClassLoaderSpi -Dappserver.platform=%PLATFORM% -Dam.home
   =%PRODUCT_HOME%\itcamdc -Ditcam61.home=%PRODUCT_HOME% -agentlib:
   am_sun_15 -DArm40.ArmTransactionFactory=com.ibm.tivoli.itcam.toolkit.arm.j2.
   transaction.Arm40TransactionFactory -DITCAMfJ2=true -DArm4EventListener.
   0=com.ibm.tivoli.itcam.dc.event.ARM4TransactionDataHandler -Dcom.ibm.tivoli.
   transperf.instr.probes.impl.was.Globals.traceLevel=0 -Dorg.omg.
   PortableInterceptor.ORBInitializerClass.com.ibm.tivoli.itcam.dc.
   orbinterpretor.Initializer -Xloggc:"E:\TOMCA5~1\DC\tomcat123-gc-log.log.
   ibmtest" -Djava.security.policy=E:\TOMCA5~1\DC\runtime\tomcat123.tivoli.us.
   abc.com.ibmtest\tivu15.cn.ibm.com.ibmtest.datacollector.policy
   ```

6. Go to the Control Panel, click System, and click the Advanced tab.
7. Click Environment Variables.
8. Under System variables, add <DC_home>\toolkit\lib\winntto the Path variable. (Replace <DC_home> with the real path for the Data Collector installation directory.)
9. Add the new variables QUALDIRand CCLOG_COMMON_DIR. Specify the values that are in catalina.bat file.
10. Restart Windows.

Appendix A. JMX reference information

The following information applies to JMX communications with J2EE servers.

J2SE JMXEnginePlugin interface

```
package com.ibm.tivoli.itcam.j2se.jmxe;

/**
 * This is interface of JMX Engine Plugin. If the J2SE application has
 * an embeded JMX server which can not return MBeanServer by MBeanServerFactory.
   findMBeanServer(null),
 * then user needs to implement this interface to return a working MBeanServer
 * instance.
 */
public interface JMXEnginePlugin {
  /**
   * The system passes necessary properties to user's implementation by this
     function,
   * for example, the PORT, USERNAME, PASSWORD to connect to JMX Server remotely.
   *
   * @param prop necessary properties to connect JMX Server
   * @throws Exception user defined initialization error
   */
   public void initialize(Properties prop) throws Exception;

  /**
   * Get MBeanServer for (un)registration of MBean
   * @return a working MBeanServer that DC can (un)register MBean
   */
   public MBeanServer getRegistrationMBeanServer();

  /**
   * This method is user's implementation to query attribute of a MBean
   * from JMX Server. There is a default implementation from JMX engine
   *
   * @param proxy - object reference
   * @param method - method name
   * @param args - method arguments
   */
  public Object invoke(Object proxy, Method method, Object[] args) throws Throwable;

  /**
   * This method is user's implementation to compose ObjectName for those
   * MBeans to be registered into JMX Server. There is a default implementation
   * from MBeanManager. The string returned by user's function will be inserted
   * before the string "Type=xxx, Name=yyy" which is returned from default function
   * in MBeanManager.
   * @param name : name of MBean.
   * @param type : type of MBean.
   * @param extraProp extra properties of MBean.
   * @return The String of ObjectName
   */
  public String buildObjectNameString(String domainName, String type, String name,
  Properties extraPrope;

  public final static String  HOST                    = J2SELocalSettings.HOST;
  public final static String  PORT                    = J2SELocalSettings.PORT;
  public final static String  USERNAME                = J2SELocalSettings.USERNAME;
  public final static String  PASSWORD                = J2SELocalSettings.PASSWORD;
}
```

J2SE JMX plug-in sample

```
package com.testware.standalone.jmx;

import java.lang.reflect.Method;
import java.util.HashMap;
import java.util.Map;
import java.util.Properties;
import javax.management.AttributeNotFoundException;
import javax.management.MBeanServer;
import javax.management.MBeanServerConnection;
import javax.management.ObjectName;
import javax.management.QueryExp;
import javax.management.remote.JMXConnector;
import javax.management.remote.JMXConnectorFactory;
import javax.management.remote.JMXServiceURL;
import com.ibm.tivoli.itcam.j2se.jmxe.JMXEnginePlugin;

public class CustomJMXEngine implements JMXEnginePlugin {
 private String hostip = "";
 private int port = 0;
 private String username = "";
 private String password = "";
 MBeanServerConnection mbsc = null;
 /**
  * @param prop All variables about jmx will be set into this properties.
     Such as host, port, username and password
  */
 public void initialize(Properties prop) throws Exception {
  this.hostip = prop.getProperty(HOST,"127.0.0.1");
  String port_s = prop.getProperty(PORT);
  try {
   this.port = Integer.parseInt(port_s);
  } catch (NumberFormatException e) {
   this.port = 0;
  }
  this.username = prop.getProperty(USERNAME);
  this.password = prop.getProperty(PASSWORD);

  if(mbsc == null)
  {
   MBeanUtils.getInstance().createMBeanServer();
   mbsc = this.getMBeanServerConnection();
  }
 }
 private MBeanServerConnection getMBeanServerConnection() throws Exception {
  // Get MBeanServerConnection
  MBeanServerConnection connection;
  try {
   // The address of the connector server
   JMXServiceURL url = new JMXServiceURL("rmi", this.hostip, this.port,
"/jndi/jmx");

   // The credentials are passed via the environment Map
   Map environment = new HashMap();
   String[] credentials = new String[]{this.username, this.password};
   environment.put(JMXConnector.CREDENTIALS, credentials);

   // Connect to the server
   JMXConnector cntor = JMXConnectorFactory.connect(url, environment);

   connection = cntor.getMBeanServerConnection();
  } catch (Exception e) {
   e.printStackTrace();
   throw e;
  }
  return connection;
```

```
 }
 /**
  * Framwork use this method to get customer's mbean server and register
some mbeans into it
  */
 public MBeanServer getRegistrationMBeanServer() {
  MBeanServer server = MBeanUtils.getInstance().getMBeanServer();
  return server;
 }
 /**
  * Proxy function to invoke method of MBean Object.
  */
 public Object invoke(Object proxy, Method method, Object[] args) throws
Throwable {
  Object returnValue = null;

  try {
   if (method.getName().equals("getDefaultDomain")) {
    returnValue = mbsc.getDefaultDomain();
   } else if (method.getName().equals("queryNames")) {
    returnValue = mbsc.queryNames((ObjectName) args[0],(QueryExp) args[1]);
   } else if (method.getName().equals("getAttribute")) {
    returnValue = mbsc.getAttribute((ObjectName) args[0],(String) args[1]);
   } else if (method.getName().equals("invoke")) {
    returnValue = mbsc.invoke((ObjectName) args[0],(String) args[1],
(Object[]) args[2],(String[]) args[3]);
   } else if (method.getName().equals("getMBeanInfo")) {
    returnValue = mbsc.getMBeanInfo((ObjectName) args[0]);
   } else if (method.getName().equals("getAttributes")) {
    returnValue = mbsc.getAttributes((ObjectName) args[0],(String[]) args[1]);
   } else {
     throw new Exception(method.getName()
     + " IS NOT IMPLEMENTED OR IS UNKNOWN");
   }
  }catch (AttributeNotFoundException e) { //ignore all attribute not found
excpetion.
  }catch (Exception re) {
   re.printStackTrace();
  }
  return returnValue;
 }
 public String buildObjectNameString(String domainName, String type, String name,
Properties extraProperties) {
  return null;
 }
}
```

Appendix B. Configure Tomcat Data Collector with Java Service Wrapper

To support Tomcat Data Collector with Java Service Wrapper, perform the following steps:

1. Follow the installation and customization guide to install and configure the Tomcat Data Collector as usual. The purpose of this step is to obtain the configuration settings defined in the *catalina.sh* by the Data Collector Configuration Tools. In the next step, we will move the configuration settings from *catalina.sh* to the wrapper configuration file *(wrapper.conf)*.
2. Using a text editor, move the configuration settings from *catalina.sh* to *wrapper.conf*. There are three types of settings to be moved:
   a. Environment settings

      In *catalina.sh*, they are defined as follows. Please remove these lines from the file. Note that some of the values should be defined differently, depending on the environment the Data Collector is running on.

      ```
      PRODUCT_HOME=/export/gqwang/tomcatdc
      export PRODUCT_HOME
      MS_HOME=%2Fopt%2FIBM%2Fitcam%2FWebSphere%2FMS
      export MS_HOME
      APPSERVER=wrapper_tomcat_server
      export APPSERVER
      NODENAME=tivsun06.cn.ibm.com
      export NODENAME
      PLATFORM=tomcat55
      export PLATFORM
      QUALDIR=tivsun06.cn.ibm.com.wrapper_tomcat_server
      export QUALDIR
      CCLOG_COMMON_DIR="/var/ibm/tivoli/common"
      export CCLOG_COMMON_DIR
      LD_LIBRARY_PATH=/export/gqwang/tomcatdc/toolkit/lib/solaris2:${LD_LIBRARY_PATH}
      export LD_LIBRARY_PATH
      ```

      When the settings are moved to *wrapper.conf*, they should be defined as follows. Note that some of the values should be defined differently, depending on the environment the Data Collector is running on.

      ```
      set.PRODUCT_HOME=/export/gqwang/tomcatdc
      set.MS_HOME=%2Fopt%2FIBM%2Fitcam%2FWebSphere%2FMS
      set.APPSERVER=wrapper_tomcat_server
      set.NODENAME=tivsun06.cn.ibm.com
      set.PLATFORM=tomcat55
      set.QUALDIR=tivsun06.cn.ibm.com.wrapper_tomcat_server
      set.CCLOG_COMMON_DIR="/var/ibm/tivoli/common"
      set.LD_LIBRARY_PATH=%PRODUCT_HOME%/toolkit/lib/solaris2:%LD_LIBRARY_PATH%
      wrapper.java.library.path.append_system_path=true
      ```

   b. JAVA options

      In *catalina.sh*, they are defined as follows. Please remove these lines from the file. Note that some of the values should be defined differently, depending on the environment the Data Collector is running on.

      ```
      JAVA_OPTS="-Xbootclasspath/p:$PRODUCT_HOME/toolkit/lib/bcm-bootstrap.jar:
        /export/gqwang/tomcatdc/toolkit/lib/jiti.jar:$PRODUCT_HOME/itcamdc/
        lib/ppe.probe-bootstrap.jar
      -Djava.rmi.server.RMIClassLoaderSpi=com.ibm.tivoli.itcam.tomcat.sdc.
        DCRMIClassLoaderSpi
      -Dam.appserver=$APPSERVER
      -Dam.nodename=$NODENAME
      ```

```
-Dappserver.platform=$PLATFORM
-Dam.home=$PRODUCT_HOME/itcamdc
-Ditcam61.home=$PRODUCT_HOME
-Xrunam_sun_15:/export/gqwang/tomcatdc/runtime/tomcat55.tivsun06.cn.
  ibm.com.wrapper_tomcat_server/jiti.properties
-Djlog.propertyFileDir.CYN=$PRODUCT_HOME/toolkit/etc
-Dcom.ibm.tivoli.itcam.toolkit.util.logging.qualDir=$QUALDIR
-Djlog.propertyFile=cynlogging.properties
-Djlog.qualDir=$NODENAME.$APPSERVER
-DArm40.ArmTransactionFactory=com.ibm.tivoli.itcam.toolkit.arm.j2.
  transaction.Arm40TransactionFactory
-DITCAMfJ2=true
-DArm4EventListener.0=com.ibm.tivoli.itcam.dc.event.
  ARM4TransactionDataHandler
-Dcom.ibm.tivoli.transperf.instr.probes.impl.was.Globals.traceLevel=0
-Dcom.ibm.tivoli.jiti.injector.IProbeInjectorManager=com.ibm.tivoli.
  itcam.toolkit.ai.bcm.bootstrap.ProbeInjectorManager
-Dorg.omg.PortableInterceptor.ORBInitializerClass.com.ibm.tivoli.itcam.
  dc.orbinterpretor.Initializer
-Dibm.common.log.dir=/var/ibm/tivoli/common
-Djlog.common.dir=/var/ibm/tivoli/common
-Djlog.qualDir=tivsun06.cn.ibm.com.wrapper_tomcat_server"
```

When the settings are moved to *wrapper.conf*, they should be defined as follows. Note that some of the values should be defined differently, depending on the environment the Data Collector is running on.

```
wrapper.java.additional.1=-Xbootclasspath/p:%PRODUCT_HOME%/toolkit/lib/
  bcm-bootstrap.jar:/export/gqwang/tomcatdc/toolkit/lib/jiti.jar:
  %PRODUCT_HOME%/itcamdc/lib/ppe.probe-bootstrap.jar
wrapper.java.additional.2=-Djava.rmi.server.RMIClassLoaderSpi=com.ibm.
  tivoli.itcam.tomcat.sdc.DCRMIClassLoaderSpi
wrapper.java.additional.3=-Dam.appserver=%APPSERVER%
wrapper.java.additional.4=-Dam.nodename=%NODENAME%
wrapper.java.additional.5=-Dappserver.platform=%PLATFORM%
wrapper.java.additional.6=-Dam.home=%PRODUCT_HOME%/itcamdc
wrapper.java.additional.7=-Ditcam61.home=%PRODUCT_HOME%
wrapper.java.additional.8=-Xrunam_sun_15:/export/gqwang/tomcatdc/runtime/
  tomcat55.tivsun06.cn.ibm.com.wrapper_tomcat_server/jiti.properties
wrapper.java.additional.9=-Djlog.propertyFileDir.CYN=%PRODUCT_HOME%/
  toolkit/etc
wrapper.java.additional.11=-Dcom.ibm.tivoli.itcam.toolkit.util.logging.
  qualDir=%QUALDIR%
wrapper.java.additional.12=-Djlog.propertyFile=cynlogging.properties
wrapper.java.additional.13=-Djlog.qualDir=%NODENAME%.%APPSERVER%
wrapper.java.additional.14=-DArm40.ArmTransactionFactory=com.ibm.tivoli.
  itcam.toolkit.arm.j2.transaction.Arm40TransactionFactory
wrapper.java.additional.15=-DITCAMfJ2=true
wrapper.java.additional.16=-DArm4EventListener.0=com.ibm.tivoli.itcam.
  dc.event.ARM4TransactionDataHandler
wrapper.java.additional.17=-Dcom.ibm.tivoli.transperf.instr.probes.impl.
  was.Globals.traceLevel=0
wrapper.java.additional.18=-Dcom.ibm.tivoli.jiti.injector.
  IProbeInjectorManager=com.ibm.tivoli.itcam.toolkit.ai.bcm.bootstrap.
  ProbeInjectorManager
wrapper.java.additional.19=-Dorg.omg.PortableInterceptor.
  ORBInitializerClass.com.ibm.tivoli.itcam.dc.orbinterpretor.Initializer
wrapper.java.additional.20=-Dibm.common.log.dir=/var/ibm/tivoli/common
wrapper.java.additional.21=-Djlog.common.dir=/var/ibm/tivoli/common
wrapper.java.additional.22=-Djlog.qualDir=tivsun06.cn.ibm.com.
  wrapper_tomcat_server
```

c. JAVA CLASSPATH

In *catalina.sh*, they are defined as follows. Please remove these lines from the file. Note that some of the values should be defined differently, depending on the environment the Data Collector is running on.

```
CLASSPATH=$PRODUCT_HOME/itcamdc/lib/ext/tomcat/loader/ppe.
  probe_tomcat.loader.jar:$CLASSPATH
CLASSPATH=$PRODUCT_HOME/toolkit/lib/ext/tk_jdbc_aspects.jar:
  $PRODUCT_HOME/toolkit/lib/ext/tk_cl_aspects.jar:$CLASSPATH
```

When the settings are moved to *wrapper.conf*, they should be defined as follows. Note that some of the values should be defined differently, depending on the environment the Data Collector is running on.

```
wrapper.java.classpath.6=%PRODUCT_HOME%/toolkit/lib/ext/tk_jdbc_aspects.jar
wrapper.java.classpath.7=%PRODUCT_HOME%/toolkit/lib/ext/tk_cl_aspects.jar
wrapper.java.classpath.10=%PRODUCT_HOME%/itcamdc/lib/ext/tomcat/loader/
  ppe.probe_tomcat.loader.jar
```

3. Restart the Tomcat server after the configuration changes.

Note: Avoid making the following mistakes when editing *wrapper.conf*:

- Repetitive sequence number. For example:

```
wrapper.java.additional.36=...
wrapper.java.additional.37=...
wrapper.java.additional.37=...
```

- Missing sequence number. For example:

```
wrapper.java.additional.35=...
wrapper.java.additional.37=...
wrapper.java.additional.38=...
```

- Double quotation marks on the *wrapper.java.additional* settings.

Appendix C. Setting up security

You can set up security for agent communication with the Managing Server.

Node Authentication

Node Authentication is the technique used to ensure that the managing server and data collectors communicate with each other in a secure manner. In Node Authentication related configuration, the Kernel, Data Collectors or Port Consolidator operate in secure mode either individually or in combination. The configuration changes are common for all the modes except that a particular component can be made to operate in a different mode by changing the property security.enabled on that particular component. You can use the following combinations:

- Managing server in secure mode and the data collector in non secure mode.
- Data collector in secure mode and the managing server in non secure mode.
- Managing server and data collector in secure or non secure mode.

Script to run if your SSL certificates have expired

All SSL certificates have an expiration time. For some certificates, the expiration time is 4 years, after which the product will not function if you have enabled Node Authentication and SSL. If this is the case, to increase the expiration time, perform the procedure at "Script to run if your SSL certificates have expired" on page 94.

Node Authentication on the Managing Server

The following procedures are Node Authentication related configuration that occurs on the Managing Server component.

Kernel-related changes

In the managing server in the $MSHOME/bin directory there is setenv.sh file that is shared by all managing server components. All changes made to the setenv.sh file apply to all managing server components. All the managing server components initialize their respective security modules based on the properties in this setenv.sh file. The installer configures all managing server components with security enabled configuration by default with the exception of kernel-related changes which are enabled by changing the .kl1 and .kl2 property files on the managing server.

In the Kernel properties file (*MS_home*/etc/kl1.properties) complete the following steps:

1. To enable a Kernel to operate in secure mode, set the following property:

   ```
   security.enabled=true
   ```

2. If you have a multiple Network Interface Card (NIC) environment or are upgrading the Managing Server from version 6.0 to version 7.1.0.2, in the Kernel properties file (*MS_home*/etc/kl1.properties), set `codebase.security.enabled=false`.

   If you have more than one instance of the Kernel, set `codebase.security.enabled=false` in kl2.properties as well.

3. Restart the Managing Server. See *IBM Tivoli Composite Application Manager for Application Diagnostics Managing Server Installation Guide*.

Data Collector custom properties file changes

The following procedure is Node Authentication related configuration that occurs by modifying the datacollector_custom.properties file.

Enabling the Data Collector to operate in secure mode

In the Data Collector custom properties file (*DC_home*/runtime/*app_server_version.node_name.server_name*/custom/datacollector_custom.properties) complete the following steps:

1. Set `security.enabled=true`
2. Restart the application server.

Node Authentication related properties in the Port Consolidator

The following procedure is Node Authentication related configuration that occurs by modifying the proxy.properties file.

In the Port Consolidator properties file (*DC_home*/itcamdc/etc/proxy.properties) complete the following steps.

1. To enable the Port Consolidator to operate in secure mode:
   `security.enabled=true`
2. Restart the application server.

See Appendix E, "Port Consolidator reference and configuration," on page 99 for instructions on configuring the Data Collector to use the Port Consolidator.

Keystore management and populating certificates

You do not have to use the following commands unless you want to create unique certificates with a new storepass and keypass. You can run keystore management on the managing server and the data collector. These commands will populate a new store with those certificates.

For populating all new keystores : there are 3 stores used by ITCAM for Application Diagnostics: CyaneaMgmtStore to run on the managing server, CyaneaDCStore to run on the data collectors, and CyaneaProxyStore to run on the data collector when you want to enable the data collector port consolidator.

CyaneaMgmtStore contains: mgmttomgmt.cer (cn=cyaneamgmt)dctomgmt.cer (cn=cyaneadc)proxytomgmt.cer (cn=cyaneaproxy)

CyaneaDCStore contains: proxytodc.cer (cn=cyaneaproxy) mgmttodc.cer (cyaneamgmt)

CyaneaProxyStore contains: mgmttoproxy.cer (cn=cyaneamgmt) dctoproxy.cer (cn=cyaneadc)

To run the keytool commands, you must be in the java/bin directory or have keytool in your PATH. This is the command with the necessary parameters:

keytool -genkey -alias *alias_name* -keyalg RSA -keysize 1024 -sigalg MD5withRSA -validity 2000 -keypass *keypass* -keystore ./*storename* -storepass *storepass* -dname "cn=cyaneamgmt, OU=CyaneaComp, O=Cyanea, L=Oakland, ST=CA, C=US"

Use the following details to create all the necessary stores and certificates:

Note: Replace "oakland1" with your custom keypass and "oakland2" with your custom storepass. Replace "CyaneaMgmtStore", "CyaneaDCStore", and "CyaneaProxyStore" with your custom store names.

```
keytool -genkey -alias mgmttomgmt -keyalg RSA -keysize 1024 -sigalg MD5withRSA
  -validity 2000 -keypass oakland1 -keystore ./CyaneaMgmtStore
  -storepass oakland2 -dname "cn=cyaneamgmt, OU=CyaneaComp, O=Cyanea, L=Oakland,
  ST=CA, C=US"

keytool -genkey -alias dctomgmt -keyalg RSA -keysize 1024 -sigalg MD5withRSA
  -validity 2000 -keypass oakland1 -keystore ./CyaneaMgmtStore -storepass oakland2
  -dname "cn=cyaneadc, OU=CyaneaComp, O=Cyanea, L=Oakland, ST=CA, C=US"

keytool -genkey -alias proxytomgmt -keyalg RSA -keysize 1024 -sigalg MD5withRSA
  -validity 2000 -keypass oakland1 -keystore ./CyaneaMgmtStore -storepass oakland2
  -dname "cn=cyaneaproxy, OU=CyaneaComp, O=Cyanea, L=Oakland, ST=CA, C=US"

keytool -genkey -alias proxytodc -keyalg RSA -keysize 1024 -sigalg MD5withRSA
  -validity 2000 -keypass oakland1 -keystore ./CyaneaDCStore -storepass oakland2
  -dname "cn=cyaneaproxy, OU=CyaneaComp, O=Cyanea, L=Oakland, ST=CA, C=US"

keytool -genkey -alias mgmttodc -keyalg RSA -keysize 1024 -sigalg MD5withRSA
  -validity 2000 -keypass oakland1 -keystore ./CyaneaDCStore
  -storepass oakland2 -dname "cn=cyaneamgmt, OU=CyaneaComp, O=Cyanea,
  L=Oakland, ST=CA, C=US"

keytool -genkey -alias mgmttoproxy -keyalg RSA -keysize 1024 -sigalg MD5withRSA
  -validity 2000 -keypass oakland1 -keystore ./CyaneaProxyStore -storepass oakland2
  -dname "cn=cyaneamgmt, OU=CyaneaComp, O=Cyanea, L=Oakland, ST=CA, C=US"

keytool -genkey -alias dctoproxy -keyalg RSA -keysize 1024 -sigalg MD5withRSA
  -validity 2000 -keypass oakland1 -keystore ./CyaneaProxyStore
  -storepass oakland2 -dname "cn=cyaneadc, OU=CyaneaComp, O=Cyanea, L=Oakland,
  ST=CA, C=US"
```

Extracting Certificates:

When you have created the three 3 Stores, extract the certificates by completing the following steps:

1. Extract all certificates from CyaneaMgmtStore by running the following commands:

   ```
   keytool -export -alias mgmttomgmt -keypass oakland1 -keystore ./CyaneaMgmtStore
     -storepass oakland2 -file mgmttomgmt.cer

   keytool -export -alias dctomgmt -keypass oakland1 -keystore ./CyaneaMgmtStore
     -storepass oakland2 -file dctomgmt.cer

   keytool -export -alias proxytomgmt -keypass oakland1 -keystore ./CyaneaMgmtStore
     -storepass oakland2 -file proxytomgmt.cer
   ```

2. Extract all certificates from CyaneaDCStore by running the following commands:

   ```
   keytool -export -alias proxytodc -keypass oakland1 -keystore ./CyaneaDCStore
     -storepass oakland2 -file proxytodc.cer

   keytool -export -alias mgmttodc -keypass oakland1 -keystore ./CyaneaDCStore
     -storepass oakland2 -file mgmttodc.cer
   ```

3. Extract all certificates from CyaneaProxyStore by running the following commands:

   ```
   keytool -export -alias mgmttoproxy -keypass oakland1
     -keystore ./CyaneaProxyStore -storepass oakland2 -file mgmttoproxy.cer

   keytool -export -alias dctoproxy -keypass oakland1
     -keystore ./CyaneaProxyStore -storepass oakland2 -file dctoproxy.cer
   ```

When you have extracted your files, copy the following certificates and Stores to the following locations:

MS_home/etc:CyaneaMgmtStore mgmttoproxy.cer mgmttomgmt.cer mgmttodc.cer

DC_home/itcamdc/etc:CyaneaDCStore CyaneaProxyStore proxytomgmt.cerproxytodc.cerdctoproxy.cer dctomgmt.cer

Configuring components to use new keystores and certificates

Configure components to use new keystores and certificates:

1. Modify *MS_home*/bin/setenv.sh. At the end of the script you will need to modify the following lines with the new keystore name, storepass, and keypass:

   ```
   KEYSTR_LOC=MS_home/etc/IBMMSStore
   KEYSTR_PASS=oakland2
   KEYSTR_KEYPASS=oakland1
   ```

2. Modify the Visualization Engine (Application Monitor) user interface with the new keystore name, storepass and keypass. Perform the following procedure:
   a. Start the Managing Server.
   b. Log into the IBM WebSphere Application Server administrative console.
   c. Click **Server > Application Servers** and select the *server_name*.
   d. In the **Configuration** tab, navigate to **Server Infrastructure: Java and Process Management > Process Definition > Additional Properties: Java Virtual Machine > Additional Properties: Custom Properties**.
   e. For the following name and value pairs, click **New**, enter the Name and Value, and click **Apply**:
      1) Set the path of the certificate to use when security is enabled for the Visualization Engine (Application Monitor) user interface:

         ```
         certificate.path=MS_home/etc/mgmttomgmt.cer
         ```

      2) Set the keystore location of the Managing Server:

         ```
         keystore.location=MS_home/etc/CyaneaMgmtStore
         ```

      3) Set the keystore password of Managing Server:

         ```
         keystore.storepass=oakland2
         ```

      4) Set the keystore key password of Managing Server:

         ```
         keystore.keypass=oakland1
         ```

      5) Set the user ID passed to the other end for authentication:

         ```
         nodeauth.userid=cyaneamgmt
         ```

   f. Restart the application server.
3. Modify *DC_home*/runtime/*app_server_version.node_name.server_name*/custom/ datacollector_custom.properties file with the new storename, storepass and keypass.
   a. Stop the instance of the application server that is being monitored by the Data Collector.
   b. Go to DC_home/runtime/app_server_version.node_name.server_name/ custom/ datacollector_custom.properties.
   c. Set the following property definitions:

      Note: All the following properties are set during the installation or at configuration time. By default you do not need to do anything. You only need to change the following properties if you have changed items that the following properties are referring to. All the keywords in angle (< >) brackets need to be replaced by the appropriate value.

      - The path of the certificate to use when communicating with the data collector. This is only needed when the data collector is operating in secure mode. The delimiter must be a semicolon on all platforms certificate.path=<AM_HOME>/etc/dctomgmt.cer;<AM_HOME>/etc/ dctoproxy.cer.

- The keystore location of the data collector keystore.location=@{AM_HOME}/etc/CyaneaDCStore.
- The keystore password of data collector server keystore.storepass=oakland94612.
- The keystore key password of data collector server keystore.keypass=oakland94612.

d. Start the instance of the application server that is monitored by the data collector for the property changes to take effect.

4. Restart the Managing Server to implement the changes made to the Managing Server and Data Collector. See *IBM Tivoli Composite Application Manager for Application Diagnostics Managing Server Installation Guide*.

Secure Socket Layer communications

On distributed platforms, ITCAM for Application Diagnostics uses the SSL security protocol for integrity and confidentiality. You have the option of configuring all monitoring components to utilize SSL for communications. The following steps describe a sample HTTP-based SSL transaction using server-side certificates:

1. The client requests a secure session with the server.
2. The server provides a certificate, its public key, and a list of its ciphers to the client.
3. The client uses the certificate to authenticate the server (verify that the server is who it claims to be).
4. The client picks the strongest common cipher and uses the server's public key to encrypt a newly-generated session key.
5. The server decrypts the session key with its private key.
6. From this point forward, the client and server use the session key to encrypt all messages.

The monitoring software uses the Java Secure Sockets Extensions (JSSE) API to create SSL sockets in Java applications.

Note: If you performed an embedded installation of the IBM WebSphere Application Server with the Managing Server, use the IBM WebSphere Application Server default key. For more information on IBM WebSphere Application Server default keys, refer to the IBM WebSphere Application Server documentation.

Password encryption and Kernel property file encryption

The amcrypto.sh script comes with the Managing Server and is present in *MS_home*/bin to encrypt the passwords related to Node Authentication and SSL.

Password encryption

To encrypt a password, complete the following steps:

1. Enter:

   ```
   amcrypto.sh -encrypt password
   ```

   The password is written to stdout.

2. Copy this encrypted password and place it in the appropriate config files.

   Currently password encryption is supported only for the following property values on both the Managing Server and Data Collectors:

   - KEYSTR_PASS and KEYSTR_KEYPASS in *MS_home*/bin/setenv.sh

- JDBC_PASSWORD in *MS_home*/bin/setenv.sh. See *ITCAM Managing Server Installation and Customization Guide* for full instructions for changing the Java Database Connectivity (JDBC) user ID and password for the database Schema user.
- keystore.storepass, keystore.keypass using the same method mentioned in the Step 2 on page 90.
- keystore.storepass and keystore.keypass in *DC_home*/runtime/*app_server_version.node_name.server_name*/custom/datacollector_custom.properties file.

3. Restart the Managing Server to activate the password encryption changes:
   a. If it is not already stopped, stop the Managing Server.
   b. Start the Managing Server.
4. Restart the VE application server.

Properties file encryption

Complete the following steps:

1. To encrypt a properties file, use:

   ```
   amcrypto.sh -encyptPropertyFile file
   ```

   The *file* is kl1.properties or kl2.properties in *MS_home*/etc. This command encrypts the given input file and stores it in a file with different name. The user can back up the existing properties file and have it replaced by the encrypted file for more security.
2. To decrypt a properties file, use:

   ```
   amcrypto.sh -decryptPropertyFile file
   ```

   The *file* is kl1.properties or kl2.properties in *MS_home*/etc. This command decrypts the given file and writes the decrypted file to another file with a different name.
3. Restart the Managing Server to activate the changes:
   a. If it is not already stopped, stop the Managing Server.
   b. Start the Managing Server.

Enabling Secure Socket Layer at the Data Collector level

To enable SSL, enable Node Authentication first (See "Node Authentication" on page 87). SSL works only with Node Authentication enabled.

Configuration with default options involves setting one property to `true` to operate the Data Collector in SSL mode:

1. In the *DC_home*/runtime/*app_server_version.node_name.server_name*/custom/datacollector_custom.properties file, set the following property to `true` by removing the comment symbol (#) in front of the property definition (by default, this property is commented out).

   ```
   comm.use.ssl.dc=true
   ```
2. Restart the application server.

Note: On the managing server only the Kernel-related changes need to be enabled other managing server components are enabled automatically.

Verifying secure communications

To verify SSL is properly configured, look for the message labeled CYND4051I in one of the following files:

Table 16. Location of the CYND4051I message

Windows	C:\Program Files\IBM\tivoli\common\CYN\logs*node_name.server_name**java_msg_log_file*. For example: `C:\Program Files\IBM\tivoli\common\CYN\logs\IBMNode01.server1\msg-dc-Ext.log`
UNIX and **Linux**	/var/ibm/tivoli/common/CYN/logs/*node_name.server_name*/*java_msg_log_file*. For example: `/var/ibm/tivoli/common/CYN/logs/IBMNode01.server1/msg-dc-Ext.log`
z/OS®	[ITCAM_CONFIG]/runtime/wasXX.node.server/logs/CYN/logs
IBM i	/QIBM/UserData/tivoli/common/CYN/logs/*node_name.server_name*/*java_msg_log_file*. For example: `/QIBM/UserData/tivoli/common/CYN/logs/IBMNode01.server1/msg-dc-Ext.log`

That message includes the text `Join Proxy Server and Kernel successfully`.

Only the CommandAgent port uses SSL. Other ports opened by the Data Collector (the ProbeController port and the Data Collector - Publish Server port do not use SSL. Therefore, when SSL is enabled, only the data on the channels connected to the CommandAgent port is encrypted.

All the data processed on the CommandAgent channel is encrypted when SSL is enabled. The data can be classified as follows:

Table 17. Classification of the data processed on the CommandAgent channel

Classification	Data
Command and control data	Configuring and unconfiguring the Data Collector
User actions related to threads	• Starting and stopping JVM threads • Changing thread priorities • Getting thread priorities and thread status • Requesting drill down information to see cookies, etc ... • Generating thread dumps • Getting thread stack traces
System information	• information • Operating system platform information • JVM information
Application information	• All the applications installed on the monitored • Application binaries and location information • Thread pool information related to JMS, JCA, JTA, Servlet, EJB, etc ... • Data source information
Performance data	All Performance Monitoring Infrastructure data
Transport data	• ORB data • SOAP ports

Table 17. Classification of the data processed on the CommandAgent channel (continued)

Classification	Data
Memory Information	• Obtaining JVM Heap Snapshot data • Performing memory leak analysis • Performing heap dump

Privacy filtering

The following procedures describe how to enable and verify privacy filtering.

Enabling privacy filtering

Privacy filtering is used to filter out SQL, cookie, and HTTP request query strings and other private data, for example drivers license numbers. When this property is set to `true`, this data is not collected by the Data Collector.

1. Stop the instance of application server that is being monitored by the Data Collector.
2. Go to *DC_home*/runtime/*app_server_version.node_name.server_name*/custom/datacollector_custom.properties.
3. Set the following property definition:

   ```
   secure.filter.on=true
   ```

4. Start the instance of application server that is being monitored by the Data Collector.

Verifying privacy filtering

The following statement is printed out to the Data Collector log when privacy filtering is properly configured:

```
Privacy Filter is On. Http Request Query String, SQL String and Http Cookie data is
not trasmitted.
```

The log file is trace-dc.log.

Script to run if your SSL certificates have expired

All SSL certificates have an expiration time. For some certificates, the expiration time is 4 years, after which the product will not function if you have enabled Node Authentication and SSL. If this is the case, to increase the expiration time, perform the following procedure:

1. Open the script located at *MS_home*/bin/security_cert.sh with a text editor. This is the content of the script:

```
#!/bin/sh

# (C) Copyright IBM Corp. 2005  All Rights Reserved.
#
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#

# Note: This script requires $JDK_HOME to be defined and it requires
# JDK_HOME/bin/keytool to be present. This keytool is available in FULL JDK
# versions and may not be available in JRE versions of the install

# PLEASE DEFINE JDK HOME

JDK_HOME=/opt/IBM/WebSphere/AppServer6/java
```

```
PATH=${JDK_HOME}/bin:$PATH

# This script generates ALL the certificates and certificate stores required for
# ITCAMfWAS Product (DC/MS/Port Consolidator). Currently it populates
# certificates with validity of 7000 days. If you feel its too high replace
# validity period to a lower number according to your needs. Please Note: once
# limit is reached, Product will stop working when NodeAuthentication/SSL is ON
# Its your responsibility to re-generate the certificates and stores.
# Please replace ALL the certificates at DC, MS and PortCosolidator level.
# Partial replacement will NOT work


keytool -genkey -alias mgmttomgmt -keyalg RSA -keysize 1024 -sigalg MD5withRSA -validity 7000
 -keypass cyanea94612 -keystore ./CyaneaMgmtStore -storepass cyanea94612 -dname
 "cn=cyaneamgmt, OU=CyaneaComp, O=Cyanea, L=Oakland, ST=CA, C=US"

keytool -genkey -alias dctomgmt -keyalg RSA -keysize 1024 -sigalg MD5withRSA -validity 7000
 -keypass cyanea94612 -keystore ./CyaneaMgmtStore -storepass cyanea94612 -dname
 "cn=cyaneadc, OU=CyaneaComp, O=Cyanea, L=Oakland, ST=CA, C=US"

keytool -genkey -alias proxytomgmt -keyalg RSA -keysize 1024 -sigalg MD5withRSA
 -validity 7000 -keypass cyanea94612 -keystore ./CyaneaMgmtStore -storepass cyanea94612
 -dname "cn=cyaneaproxy, OU=CyaneaComp, O=Cyanea, L=Oakland, ST=CA, C=US"

keytool -genkey -alias proxytodc -keyalg RSA -keysize 1024 -sigalg MD5withRSA
 -validity 7000 -keypass oakland94612 -keystore ./CyaneaDCStore -storepass oakland94612
 -dname "cn=cyaneaproxy, OU=CyaneaComp, O=Cyanea, L=Oakland, ST=CA, C=US"

keytool -genkey -alias mgmttodc -keyalg RSA -keysize 1024 -sigalg MD5withRSA
 -validity 7000 -keypass oakland94612 -keystore ./CyaneaDCStore -storepass oakland94612
 -dname "cn=cyaneamgmt, OU=CyaneaComp, O=Cyanea, L=Oakland, ST=CA, C=US"

keytool -genkey -alias mgmttoproxy -keyalg RSA -keysize 1024 -sigalg MD5withRSA
 -validity 7000 -keypass oakland94612 -keystore ./CyaneaProxyStore -storepass oakland94612
 -dname "cn=cyaneamgmt, OU=CyaneaComp, O=Cyanea, L=Oakland, ST=CA, C=US"

keytool -genkey -alias dctoproxy -keyalg RSA -keysize 1024 -sigalg MD5withRSA
 -validity 7000 -keypass oakland94612 -keystore ./CyaneaProxyStore -storepass oakland94612
 -dname "cn=cyaneadc, OU=CyaneaComp, O=Cyanea, L=Oakland, ST=CA, C=US"

keytool -export -alias mgmttomgmt -keypass cyanea94612 -keystore ./CyaneaMgmtStore
 -storepass cyanea94612 -file mgmttomgmt.cer

keytool -export -alias dctomgmt -keypass cyanea94612 -keystore ./CyaneaMgmtStore
 -storepass cyanea94612 -file dctomgmt.cer

keytool -export -alias proxytomgmt -keypass cyanea94612 -keystore ./CyaneaMgmtStore
 -storepass cyanea94612 -file proxytomgmt.cer

keytool -export -alias proxytodc -keypass oakland94612 -keystore ./CyaneaDCStore -storepass
 oakland94612 -file proxytodc.cer

keytool -export -alias mgmttodc -keypass oakland94612 -keystore ./CyaneaDCStore -storepass
 oakland94612 -file mgmttodc.cer

keytool -export -alias mgmttoproxy -keypass oakland94612 -keystore ./CyaneaProxyStore
 -storepass oakland94612 -file mgmttoproxy.cer

keytool -export -alias dctoproxy -keypass oakland94612 -keystore ./CyaneaProxyStore
 -storepass oakland94612 -file dctoproxy.cer

cp ./CyaneaMgmtStore ./CyaneaMgmtStore_Comm
cp ./CyaneaDCStore ./CyaneaDCStore_Comm
cp ./CyaneaProxyStore ./CyaneaProxyStore_Comm

keytool -keystore ./CyaneaMgmtStore_Comm -storepass cyanea94612 -import -alias mgmttodc
```

```
 -file ./mgmttodc.cer

keytool -keystore ./CyaneaMgmtStore_Comm -storepass cyanea94612 -import -alias mgmttoproxy
 -file ./mgmttoproxy.cer

keytool -keystore ./CyaneaDCStore_Comm -storepass oakland94612 -import -alias dctomgmt
 -file ./dctomgmt.cer

keytool -keystore ./CyaneaDCStore_Comm -storepass oakland94612 -import -alias dctoproxy
  -file ./dctoproxy.cer

keytool -keystore ./CyaneaProxyStore_Comm -storepass oakland94612 -import -alias proxytodc
 -file ./proxytodc.cer

keytool -keystore ./CyaneaProxyStore_Comm -storepass oakland94612 -import -alias proxytomgmt
  -file ./proxytomgmt.cer
```

2. Specify the path for the location of the Java home directory for the JDK_HOME parameter. For example,

   ```
   JDK_HOME=D:\IBM\AppServer\java
   ```

3. If the increase in expiration time to 20 years (7000 days) is too much, modify the script. Change the value of `-validity 7000` to a lower number of days, in all instances it occurs in the script. For example, change all instances of `-validity 7000` to `-validity 3500`.
4. Save the changes and run the script.

Settings for the Data Collector if Java 2 security is enabled

By default, Data Collector configuration enables Java 2 security on the application server, and sets a permissive policy. This policy ensures that the Data Collector can run properly, and provides no other security protection. If you need a more restrictive policy, perform the following procedure to ensure that the policy becomes active and the Data Collector can still work properly.

The Data Collector sets the Java security policy file location for all monitored application server instances (`java.security.policy` system property) to *DC_home*`/itcamdc/etc/datacollector.policy`. You must edit this file in the following way:

- Remove all existing content.
- Copy the sample security policy for the Data Collector from the file *DC_home*`/itcamdc/etc/datacollector.security.policy`.
- If ITCAM for Transactions is installed on the server, add a grant statement for the ITCAM for Transactions code base to the security policy file. Follow the model for the grant statements provided in the sample datacollector.security.policy file, but use the ITCAM for Transactions installation root directory in the codeBase statement.
- Add your required security policy settings.

Save the file, and create a backup copy.

Attention: Each time you configure or reconfigure the Data Collector for an application server instance, the file *DC_home*`/itcamdc/etc/datacollector.policy` might be overwritten. To ensure that your security policy remains active, restore this file from the backup copy after configuring or reconfiguring the Data Collector for any application server instance.

Appendix D. Using regular expressions

Regular expressions are sets of symbols and characters that are used to match patterns of text. You can use regular expressions to search specific IP addresses across your Web environment. Regular expressions also enable you to search a simple, fixed URI or a complex URI pattern that matches one or more groups of transactions.

Regular expression library

An extensive library of regular expression characters and operators is available for your URI filters and IP address specifications. The International Components for Unicode (ICU) open-source development project provides this library for your use. The next section provides the most frequently used expressions for this product. However, you can refer to the following Web page for a full description of the ICU regular expression library and an explanation of how to use the characters and operators for complex expressions: http://oss.software.ibm.com/icu/userguide/regexp.html

Frequently used regular expressions

The following list highlights characters and operators most frequently used in regular expressions:

\ Quotes the character that follows it, which treats that character as a literal character or operator (not a regular expression). When you want the following characters to be treated as literal, you must precede them with a backslash:

```
* ? + [ ( ) { } ^ $ | \ . /
```

In other words, use a backslash followed by a forward slash (\/) to include a forward slash in a URI filter. Use a backslash followed by a period (\.) to include a period in a URI filter.

Example: to specify the URI pattern `http://www.ibm.com/`, use the following regular expression:

```
http:\/\/www\.ibm\.com\/
```

To specify all URIs that begin with `http://www.ibm.com/`, use the following regular expression:

```
http:\/\/www\.ibm\.com\/.*
```

. Matches any one character.

Example: to match both `ibm2` and `ibm3` within a string, use `ibm.` such as in the following example: **`http:\/\/www\.ibm.\.com\/`**

(?: ...)

Non-capturing parentheses. Groups the included pattern, but does not provide capturing of matching text. Somewhat more efficient than capturing parentheses.

Example: you can use the non-capturing parenthesis to group expressions to form more complicated regular expressions. To match a URI that starts

with one of the following URLs: `http://www.ibm.com/marketing/` or `http://www.ibm.com/sales/`, you would do a grouping with a pipe sign (`|`) (represents *or*):

`http://www.ibm.com/(?:marketing)|(?:sales)/`

* Matches the preceding element zero or more times. You must quote this character.

Example: the expression, **`ca*t`**, matches `cat`, `caat`, `ct`, and `caaaaat`. The term `cabt`, would not return as a match.

Specifying exclusions with the bang (!) operator (Quality of Service listening policies only)

Note: This section applies to the entry of URI and client IP filters for Quality of Service listening policies only.

You can use an exclamation point (**!**), also called the *bang* operator, to filter out transactions that might match the regular expressions already entered, but that are not to be considered valid transactions for this listening policy. These exclusions are considered negative filters. You can enter these exclusions as additional URI or client IP filters. The formatting of these additional filters is as follows:

URI Filter Exclusions

Use only fixed strings. For example, you can use the following strings:

!http://www.ibm.com/
!http://www.ibm.com/hr/index.html
!http://www.ibm.com/it/errorpage.html

Client IP Exclusions

The following are valid:

!*.24.45.46
!12.*.45.56
!12.24.*.56
!12.24.45.*
!12.24.45.56

You can replace any "octet" (there are four in an IP address: octet . octet . octet . octet) with a wildcard (*). Note that this is not the regular expression wildcard (.*) from the positive filters.

Appendix E. Port Consolidator reference and configuration

The Port Consolidator is used to reduce network resources. It is used on the Data Collector to limit the number of ports used by the Data Collector when communicating with the Managing Server. The Port Consolidator only consolidates the traffic in one direction: from the Managing Server to the Data Collector. All traffic from the Managing Server to the Data Collector will be routed through the Port Consolidator. However, the traffic from the Data Collector to the Managing Server is direct.

Note: Typically, all Data Collectors and Port Consolidators are installed on the same physical computer. However, it is possible to run the Port Consolidator on a different computer. Contact IBM Software Support for setup assistance in this case.

Configuring a Data Collector to use the Port Consolidator

If you have a firewall, you can avoid allocation of an excessive number of ports in the firewall for multiple Data Collectors by configuring and using the Port Consolidator.

Perform the following procedure to configure a Data Collector to use the Port Consolidator:

1. Edit the *DC_home*/runtime/*app_server_version.node_name.server_name*/custom/datacollector_custom.properties file. Add the following lines to the end of the file:

   proxy.host=*IP_address*

   This is usually the same IP address as the Data Collector computer, but it could be different in a multiple IP or virtual host scenario. In any case, specify the same IP address as the one specified in the am.socket.bindip property in *DC_home*/itcamdc/etc/proxy.properties.

   proxy.port=*port*

   This is usually 8800. In any case, specify the same port specified in the PROXY_PORT property in *DC_home*/itcamdc/bin/proxyserverctrl_*.

   Note:

   a. Do not use the loopback address for the IP address. Use a valid IP address for the local system.

   b. proxy.port must match the port number for PROXY_PORT that is specified in the startup script you run in Step 4.
2. Restart the instance of the application server that is being monitored by the Data Collector.
3. From a command prompt, move to the directory *DC_home*/itcamdc/bin.
4. Start the Port Consolidator using one of the following commands:

Table 18. Command for starting the Port Consolidator

Windows	proxyserverctrl_j2ee.bat start
UNIX and **Linux**	./proxyserverctrl_j2ee.sh start

Do not close the command prompt window.

Note: The value for `PROXY_PORT` that is specified in the script must match the value that you specified for `proxy.port` in Step 1 on page 99.

5. Open the Self-Diagnosis page of the Visualization Engine (Application Monitor) user interface, and check to see that the following components are listed:
   - `COMMANDAGENTPROXY`
   - `KERNELPROXY`
   - `PROBECONTROLLERPROXY`
6. Verify that the Data Collector is using the Port Consolidator:
   a. Look for the message labeled CYND4051I in one of the following files:

Table 19. Location of the CYND4051I message

Windows	*DC_home*\logs\CYN\logs*node_name.server_name**java_msg_log_file*. For example: C:\IBM\ITM\TMAITM6\wasdc\7.1.0.2\logs\CYN\logs\tivx44Node02.server1\msg-dc-ParentLast.log
UNIX and **Linux**	*DC_home*/logs/CYN/logs/*node_name.server_name*/*java_msg_log_file*. For example: /opt/IBM/AD7101_0505/li6263/yn/wasdc/7.1.0.2/logs/CYN/logs/tivx44Node02.server1/msg-dc-ParentLast.log

That message includes the text `Join Proxy Server and Kernel successfully`.

   b. From a new command prompt, move to the directory *DC_home*`/itcamdc/bin`, and enter one of the following commands:

Table 20. Entering the proxyserverctrl_j2ee command

Windows	`proxyserverctrl_j2ee.bat list`
UNIX and **Linux**	`./proxyserverctrl_j2ee.sh list`

You will see the Data Collector listed as one Service type, `PPECONTROLLER`. Keep this command prompt window open for future use.

7. Verify the Data Collector connection to the Port Consolidator (again) by entering one of the following commands:

Table 21. Entering the proxyserverctrl_j2ee command

Windows	`proxyserverctrl_j2ee.bat list`
UNIX and **Linux**	`./proxyserverctrl_j2ee.sh list`

You will now see the Data Collector listed as two Service types, PPECONTROLLER and PPEPROBE.

The Data Collector is configured to use the Port Consolidator.

Reconfiguring the Data Collector to bypass the Port Consolidator

If after configuring the Data Collector to use the Port Consolidator, you want the Data Collector to bypass the Port Consolidator, perform the following procedure:

1. Unconfigure the Data Collector in the Visualization Engine (Application Monitor) user interface:

a. Start the Managing Server.
b. From the top navigation, click **Administration > Server Management > Data Collector Configuration**.

   The Data Collector Management page opens.
c. Go to the Configured Data Collectors at the top of the page.
d. To unconfigure the Data Collector, select the check box that is next to the Data Collector, and click **Apply**.

The unconfigured Data Collector is added to the Unconfigured Data Collectors page.

Notes:

a. If the data collection has reports associated with it, you are prompted to delete those reports before unconfiguring the Data Collector.

2. Stop the Port Consolidator. From a command prompt, enter one of the following values:

Table 22. Entering the proxyserverctrl_j2ee command

Windows	`proxyserverctrl_j2ee.bat stop`
UNIX and **Linux**	`./proxyserverctrl_j2ee.sh stop`

3. Verify that the Port Consolidator is stopped by entering one of the following commands:

Table 23. Entering the proxyserverctrl_j2ee command

Windows	`proxyserverctrl_j2ee.bat list`
UNIX and **Linux**	`./proxyserverctrl_j2ee.sh list`

You will now see the message `KERNELPROXY is down`.

4. Reconfigure the Data Collector to bypass the Port Consolidator:
   a. Stop the application server.
   b. Edit the *DC_home*`/runtime/`*app_server_version.node_name.server_name*`/custom/datacollector_custom.properties` file. Remove the following lines from the end of the file:

      `proxy.host=`*IP address of Data Collector*

      `proxy.port=`*port*
   c. Check for the same lines in the *DC_home*`/runtime/`*appserver_version.node_name.server_name*`/`*appserver_version.node_name.server_name*`.datacollector.properties` file; if they are present, remove them.
   d. Restart the instance of the application server that is being monitored by the Data Collector.
5. Check the configuration of your Data Collector. In the Visualization Engine (Application Monitor) user interface, click **Administration > Server Management > Data Collector Configuration**.

   The Data Collector will be listed. However, it will be showing as unavailable.
6. View **Unconfigured Data Collectors**.

   Your Data Collector will be listed.

Appendix F. Glossary

application server
Software in used in an Internet environment that hosts a variety of language systems used to program database queries and general business processing.

Command line
Unix or Linux prompt line entered to carry out a certain function.

command file
File containing command prompts to launch an application. Usually terminates with the extension .cmd or .bat.

command syntax
The pattern in which command line should be written.

Configuration Tool
Component of the ITCAM for J2EE Data Collector, the tool guides users through the process of configuring and also unconfiguring the Data Collector.

Data Collector
ITCAM for J2EE product that collects data from the Managing Server for analysis and configuration.

DOS command prompt
Program in Windows by which users may enter command lines.

host Computer with a specific application or software environment installed.

J2EE Java 2 Platform, Enterprise Edition. An environment for developing and deploying multi-tier enterprise applications. J2EE simplifies development of enterprise applications by basing them on standard, modular components; it comprises a set of services, application programming interfaces (APIs), and protocols that provide the necessary functions for developing multi-tiered, Web-based applications.

Java virtual machine
Java virtual machine, or JVM, converts the Java intermediate language into machine language and then executes it.

Managing Server
The server that manages information collected on different Data Collectors.

Managing Server instance
An instance of the Managing Server.

monitored data
Data on the Managing Server that is configured for data collection. You may see this data through your DC interface agent.

product license
Terms and conditions of the product's usage.

response file
Text file containing variables and parameters required for an installation of the ITCAM for J2EE Data Collector.

setup file
File containing the installation commands for the Data Collector.

silent installation
Installation process that does not show messages or windows during the process. Parameter definitions are specified in a text file that the installation runs from.

startup script
Command lines necessary to launch the application server.

text editor
Notepad or WordPad, an editor in which to alter or write rich text documents.

unconfiguration
Process of deselecting Managing Server instances for data collection.

Appendix G. Accessibility

Accessibility features help users with physical disabilities, such as restricted mobility or limited vision, to use software products successfully.

The accessibility features in the product enable users to:

- Use assistive technologies, such as screen reader software and digital speech synthesizers, to hear what is displayed on the screen. Consult the product documentation of the assistive technology for details on using the technology with this product.
- Perform tasks with the software using only the keyboard.

General Navigation

Each page has four main sections:

- Headerbar
- Toolbar
- Main tabs
- Content

Each page has navigation points for screen readers. The following navigation points are all H1:

- Title bar
- Main tabs
- Main form
- Section labels
- Table labels

Menu Navigation

You use the Go To menu at the top of the screen to navigate to any of the applications that you have access to. The Go To menu is a cascading menu that is three levels deep at its deepest point. The following instructions describe how to get started with JAWS:

1. To get to the Go To menu press Alt+G.
2. When you open the menu, JAWS reads the first application in the menu. If JAWS does not begin to read the entry, restart the screen reader.
3. Navigate the list of applications in the menus by using the arrow keys.
4. JAWS indicates if a menu item has submenus. To get to a submenu, press the right arrow or enter.
5. Press the left arrow to move up a level in the hierarchy. If you press the left arrow at the highest level of the Go To menu, you leave the menu completely.
6. Press the Enter key to enter an application.

Accessibility help

The Accessibility Help panels provide details on general navigation, menu navigation, and hot keys. Click **Accessibility Help** from the toolbar of the product to access the help panels.

Screen reader setting

The product contains a screen reader flag. When you turn on the screen reader flag, the user interface is optimized to work with JAWS for Windows®. You use the **User** tab in the Users application to turn on the screen reader flag.

Keyboard shortcuts

You can navigate within the applications by using a combination of keys.

Accessible reports

To use the accessibility tools to read reports, you must access the reports in Microsoft Excel. In the reports applications, select the **Run Reports** option in the **Select Action** menu. With this option, you can email an .xls file version of a report to yourself at a scheduled time.

IBM and accessibility

For more information about the commitment that IBM has to accessibility, see the IBM Human Ability and Accessibility Center. The IBM Human Ability and Accessibility Center is at the following web address: http://www.ibm.com/able

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at http://www.ibm.com/us/en/legal/copytrade.shtml.

Adobe is either a registered trademark or a trademark of Adobe Systems Incorporated in the United States, other countries, or both.

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

Intel is a trademark or registered trademark of Intel Corporation or its subsidiaries in the United States and other countries.

Microsoft and Windows 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.

Java and all Java-based trademarks are trademarks or registered trademarks of Oracle and/or its affiliates.

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

Notices

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

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, 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 IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY 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 or changes in the product(s) 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 want 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 might 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 might vary significantly. Some measurements might 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 might have been estimated through extrapolation. Actual results might 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.

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.

Privacy policy considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable

information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session cookies that collect each user's user name for purposes of session management, authentication, and single sign-on configuration. These cookies cannot be disabled.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, See IBM's Privacy Policy at http://www.ibm.com/privacy and IBM's Online Privacy Statement at http://www.ibm.com/privacy/details the section entitled "Cookies, Web Beacons and Other Technologies" and the "IBM Software Products and Software-as-a-Service Privacy Statement" at http://www.ibm.com/software/info/product-privacy.

Printed in USA

SC27-6228-00