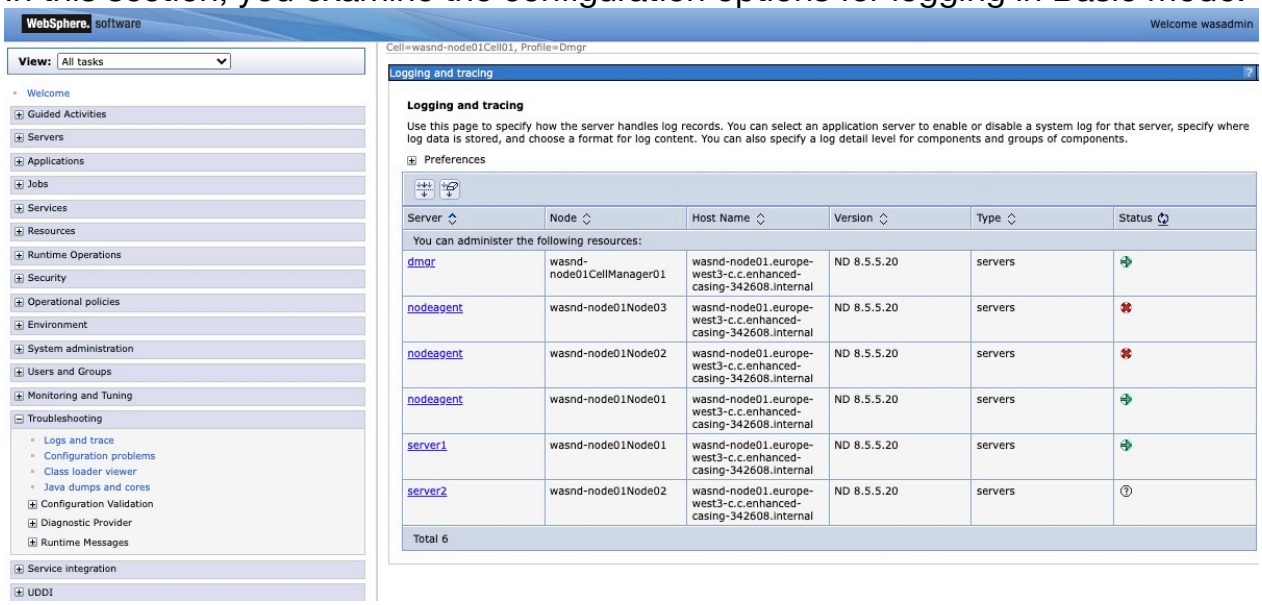
Lab 7 – Problem Determination

At the end of this exercise, you should be able to:

- Use the administrative console to configure and view log data
- Enable a server to use HPEL
- Enable tracing on application server components
- Use the HPEL Log Viewer to examine log and trace data
- Enable verbose garbage collection for an application server
- Enable memory leak detection for an application server
- Describe how IBM Support Assistant tools can be used to analyze JVM memory dumps

Section 1: Working with log files of the application server

In this section, you examine the configuration options for logging in Basic mode.



Cell=wasnd-node01Cell01, Profile=Dmgr

Logging and tracing

Logging and tracing > **server1**

It is recommended that you switch to High Performance Extensible Logging (HPEL) if you have no existing procedures that prevent you from taking advantage of it.

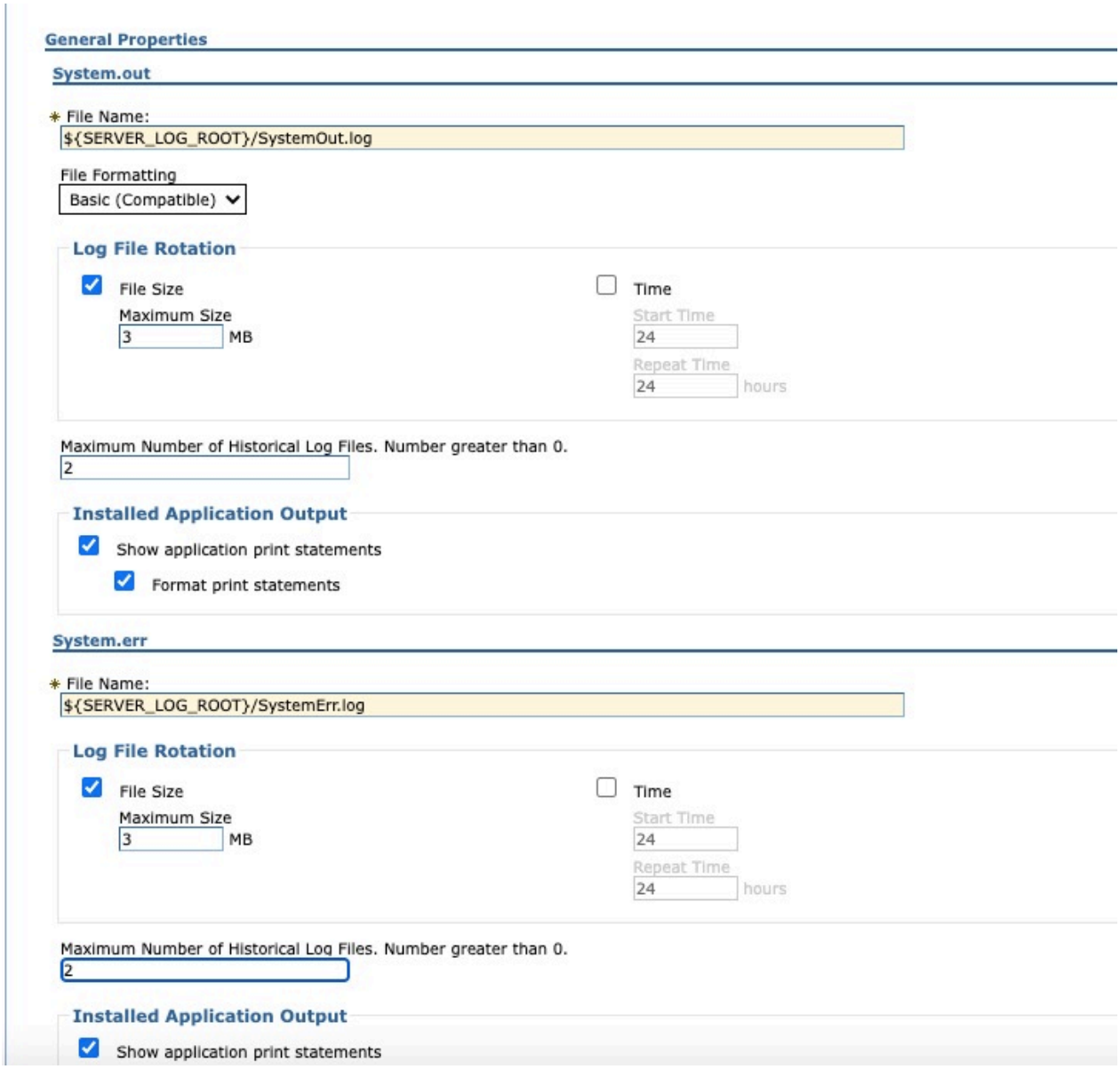
Switch to HPEL Mode

(Advised for most installations)

Use this page to select a system log to configure, or to specify a log detail level for components and groups of components. Use log levels to control which events are processed by Java logging.

General Properties

- Diagnostic Trace
- JVM Logs
- Process Logs
- IBM Service Logs
- Change log detail levels
- NCSA access and HTTP error logging


General Properties
System.out
* File Name:
${SERVER_LOG_ROOT}/SystemOut.log
File Formatting
Basic (Compatible)
Log File Rotation
File Size
Maximum Size
3
MB
Time
Start Time
24
Repeat Time
24
hours
Maximum Number of Historical Log Files. Number greater than 0.
2
Installed Application Output
Show application print statements
Format print statements
System.err
* File Name:
${SERVER_LOG_ROOT}/SystemErr.log
Log File Rotation
File Size
Maximum Size
3
MB
Time
Start Time
24
Repeat Time
24
hours
Maximum Number of Historical Log Files. Number greater than 0.
2
Installed Application Output
Show application print statements

Logging and tracing

Logging and tracing > **server1** > **JVM Logs**

Use this page to view and modify the settings for the Java virtual machine (JVM) System.out and System.err logs for a managed process. The JVM logs are created by redirecting the System.out and System.err streams of the JVM to independent log files. The System.out log is used to monitor the health of the running application server. The System.err log contains exception stack trace information that is used to perform problem analysis. One set of JVM logs exists for each application server and all of its applications. JVM logs are also created for the deployment manager and each node manager. Changes on the Configuration panel apply when the server is restarted. Changes on the Runtime panel apply immediately.

Configuration | Runtime

General Properties

System.out

File Name:
/ibm/profiles/profile1/logs/server1/SystemOut.log View

System.err

File Name:
/ibm/profiles/profile1/logs/server1/SystemErr.log View

Back

Logging and tracing

Logging and tracing > **server1** > **Log File**

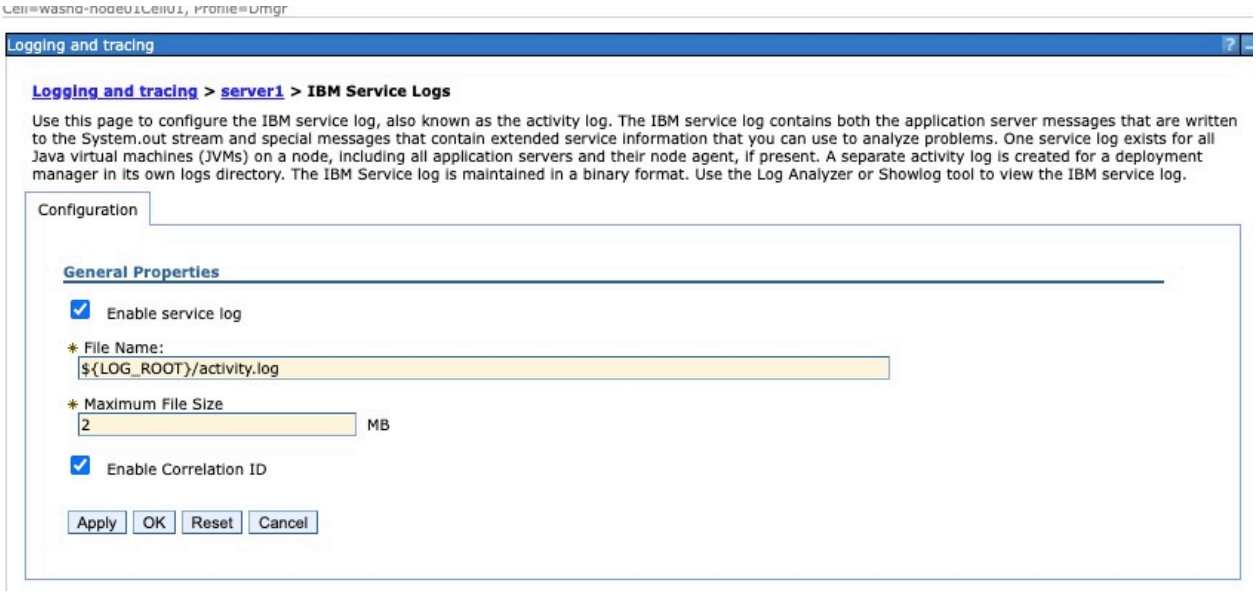
Display the contents of the given file.

Total: 1455, Filtered total: 151

Retrieve Lines (eg. 250-600)
250-400 Refresh

Log File

```
[3/4/22 16:36:45:325 UTC] 00000073 WASNameSpaceB I   CNTR0167I: The server is binding the com.ibm.websphere.ejbquery.QueryLocalHome interface of the Query enterprise bean
[3/4/22 16:36:45:327 UTC] 00000073 AbstractEJBRu I   CNTR0167I: The server is binding the com.ibm.websphere.ejbquery.QueryLocalHome interface of the Query enterprise bean
[3/4/22 16:36:45:335 UTC] 00000073 SharedEJBRunt I   WSVR0057I: EJB jar started: querybean.jar
[3/4/22 16:36:45:380 UTC] 00000074 BeansDeployer I BeansDeployer deploy All injection points were validated successfully.
[3/4/22 16:36:45:384 UTC] 00000074 WebContainerL I WebContainerLifecycle startApplication OpenWebBeans Container has started, it took [96] ms.
[3/4/22 16:36:45:385 UTC] 00000074 webcontainer  I com.ibm.ws.webcontainer.VirtualHostImpl addWebApplication SRVE0250I: Web Module Default Web Application has been bound t
[3/4/22 16:36:45:409 UTC] 00000075 ApplicationMg A   WSVR0221I: Application started: ibmasyncrsp
[3/4/22 16:36:45:410 UTC] 00000075 CompositionUn A   WSVR0191I: Composition unit WebSphere:cuname=ibmasyncrsp in BLA WebSphere:blaname=ibmasyncrsp started.
[3/4/22 16:36:45:412 UTC] 00000075 CompositionUn A   WSVR0190I: Starting composition unit WebSphere:cuname=pbw-ear in BLA WebSphere:blaname=pbw-ear.
[3/4/22 16:36:45:417 UTC] 00000073 ApplicationMg A   WSVR0221I: Application started: query
[3/4/22 16:36:45:418 UTC] 00000073 CompositionUn A   WSVR0191I: Composition unit WebSphere:cuname=query in BLA WebSphere:blaname=query started.
[3/4/22 16:36:45:418 UTC] 00000073 CompositionUn A   WSVR0190I: Starting composition unit WebSphere:cuname=ivtApp in BLA WebSphere:blaname=ivtApp.
[3/4/22 16:36:45:420 UTC] 00000074 ApplicationMg A   WSVR0221I: Application started: DefaultApplication
[3/4/22 16:36:45:420 UTC] 00000074 CompositionUn A   WSVR0191I: Composition unit WebSphere:cuname=DefaultApplication in BLA WebSphere:blaname=DefaultApplication started.
[3/4/22 16:36:45:442 UTC] 00000075 ApplicationMg A   WSVR0200I: Starting application: pbw-ear
[3/4/22 16:36:45:443 UTC] 00000075 ApplicationMg A   WSVR0204I: Application: pbw-ear  Application build level: Unknown
[3/4/22 16:36:45:456 UTC] 00000073 ApplicationMg A   WSVR0200I: Starting application: ivtApp
[3/4/22 16:36:45:457 UTC] 00000073 ApplicationMg A   WSVR0203I: Application: ivtApp  Application build level: UNKNOWN [UNKNOWN]
[3/4/22 16:36:45:501 UTC] 00000073 SharedEJBRunt I   WSVR0037I: Starting EJB jar: ivtEJB.jar
[3/4/22 16:36:45:509 UTC] 00000073 WASNameSpaceB I   CNTR0167I: The server is binding the com.ibm.websphere.ivt.ivtEJB.ivtEJBObject interface of the ivtEJBObject enterpris
[3/4/22 16:36:45:510 UTC] 00000073 WASNameSpaceB I   CNTR0167I: The server is binding the com.ibm.websphere.ivt.ivtEJB.ivtEJBObject interface of the ivtEJBObject enterpris
[3/4/22 16:36:45:511 UTC] 00000073 AbstractEJBRu I   CNTR0167I: The server is binding the com.ibm.websphere.ivt.ivtEJB.ivtEJBObject interface of the ivtEJBObject enterpris
[3/4/22 16:36:45:514 UTC] 00000073 SharedEJBRunt I   WSVR0057I: EJB jar started: ivtEJB.jar
[3/4/22 16:36:45:531 UTC] 00000073 webapp        I com.ibm.ws.webcontainer.webapp.WebGroupImpl WebGroup SRVE0169I: Loading Web Module: ivt_app.war.
[3/4/22 16:36:45:546 UTC] 00000073 WASSessionCor I SessionContextRegistry getSessionContext SESN0176I: Will create a new session context for application key default_host/i
[3/4/22 16:36:45:548 UTC] 00000073 WASSession    I MTMStore setDRSMode SESN0188I: Memory To Memory mode for application default_host/ivt is BOTH
[3/4/22 16:36:45:549 UTC] 00000073 WASSession    I MTMStore setDRSMode SESN0188I: Memory To Memory mode for application default_hostivtApp is BOTH
[3/4/22 16:36:45:560 UTC] 00000073 webcontainer  I com.ibm.ws.webcontainer.VirtualHostImpl addWebApplication SRVE0250I: Web Module null has been bound to default_host[*:9
[3/4/22 16:36:45:576 UTC] 00000073 ApplicationMg A   WSVR0221I: Application started: ivtApp
[3/4/22 16:36:45:576 UTC] 00000073 CompositionUn A   WSVR0191I: Composition unit WebSphere:cuname=ivtApp in BLA WebSphere:blaname=ivtApp started.
[3/4/22 16:36:45:607 UTC] 00000075 InternalGener I   DSRA8225I: DataSource JNDI name : jdbc/PlantsByWebSphereDataSource
[3/4/22 16:36:45:608 UTC] 00000075 InternalGener I   DSRA8203I: Database product name : Apache Derby
[3/4/22 16:36:45:608 UTC] 00000075 InternalGener I   DSRA8204I: Database product version : 10.8.3.3 - (10.12 CVE fixes)
[3/4/22 16:36:45:608 UTC] 00000075 InternalGener I   DSRA8205I: JDBC driver name  : Apache Derby Embedded JDBC Driver
[3/4/22 16:36:45:609 UTC] 00000075 InternalGener I   DSRA8206I: JDBC driver version  : 10.8.3.3 - (10.12 CVE fixes)
[3/4/22 16:36:45:609 UTC] 00000075 InternalGener I   DSRA8218I: JDBC driver specification level  : 4.0
[3/4/22 16:36:45:947 UTC] 00000075 PluginLoader  I PluginLoader loadEJBMetadata Adding OpenWebBeansPlugin : [OpenWebBeansJsfPlugin]
[3/4/22 16:36:45:948 UTC] 00000075 PluginLoader  I PluginLoader loadEJBMetadata Adding OpenWebBeansPlugin : [WSWebPlugin]
[3/4/22 16:36:45:949 UTC] 00000075 PluginLoader  I PluginLoader loadEJBMetadata Adding OpenWebBeansPlugin : [WSEjbPlugin]
[3/4/22 16:36:45:950 UTC] 00000075 PluginLoader  I PluginLoader loadEJBMetadata Adding OpenWebBeansPlugin : [ContextsServicePlugin]
[3/4/22 16:36:45:950 UTC] 00000075 PluginLoader  I PluginLoader loadEJBMetadata Adding OpenWebBeansPlugin : [OpenWebBeansEjbLCAPluginImpl]
```



Information

The IBM service log contains both the application server messages that are written to the `System.out` stream and special messages that contain extended service information that you can use to analyze problems. One service log exists for all Java virtual machines (JVMs) on a node, including all application servers and their node agent, if present. A separate activity log is created for a deployment manager in its own logs directory. The IBM Service log is maintained in a binary format. Use the Log Analyzer or Showlog tool to view the IBM service log.

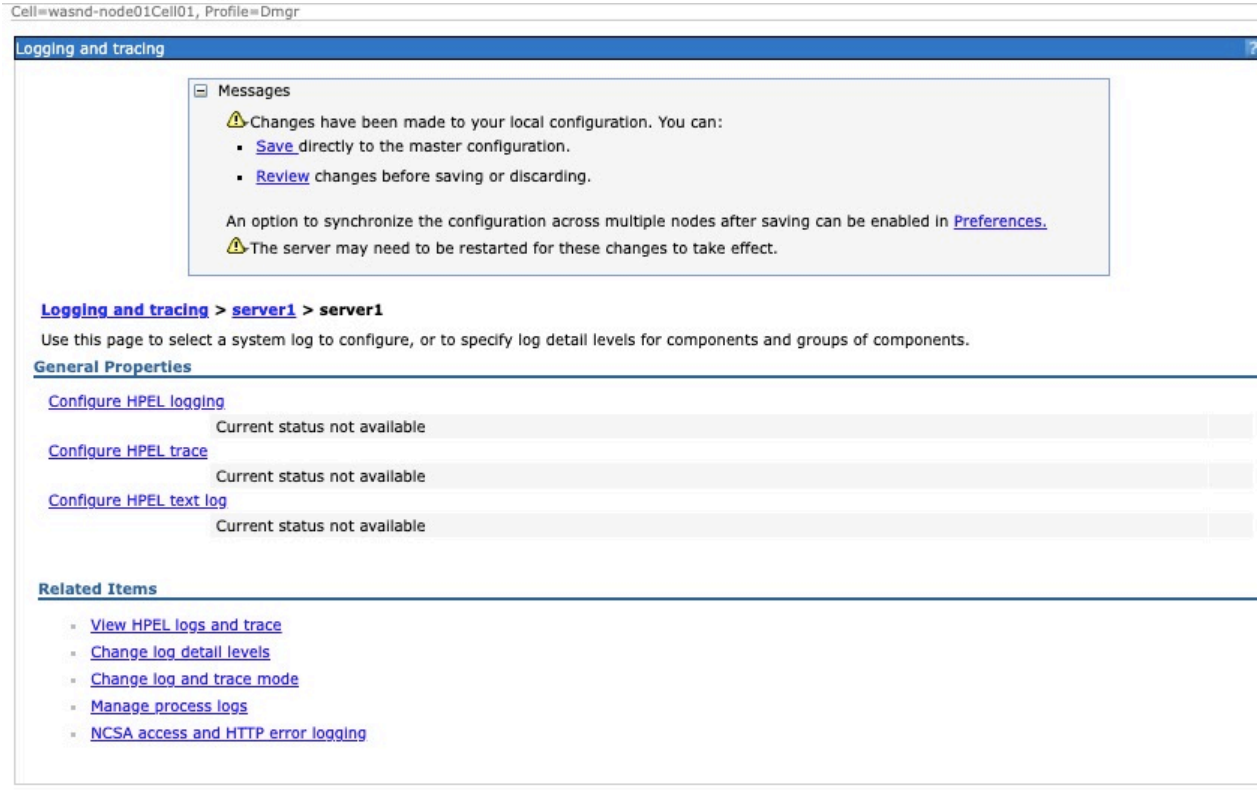
Section 2: Set up and configure HPEL

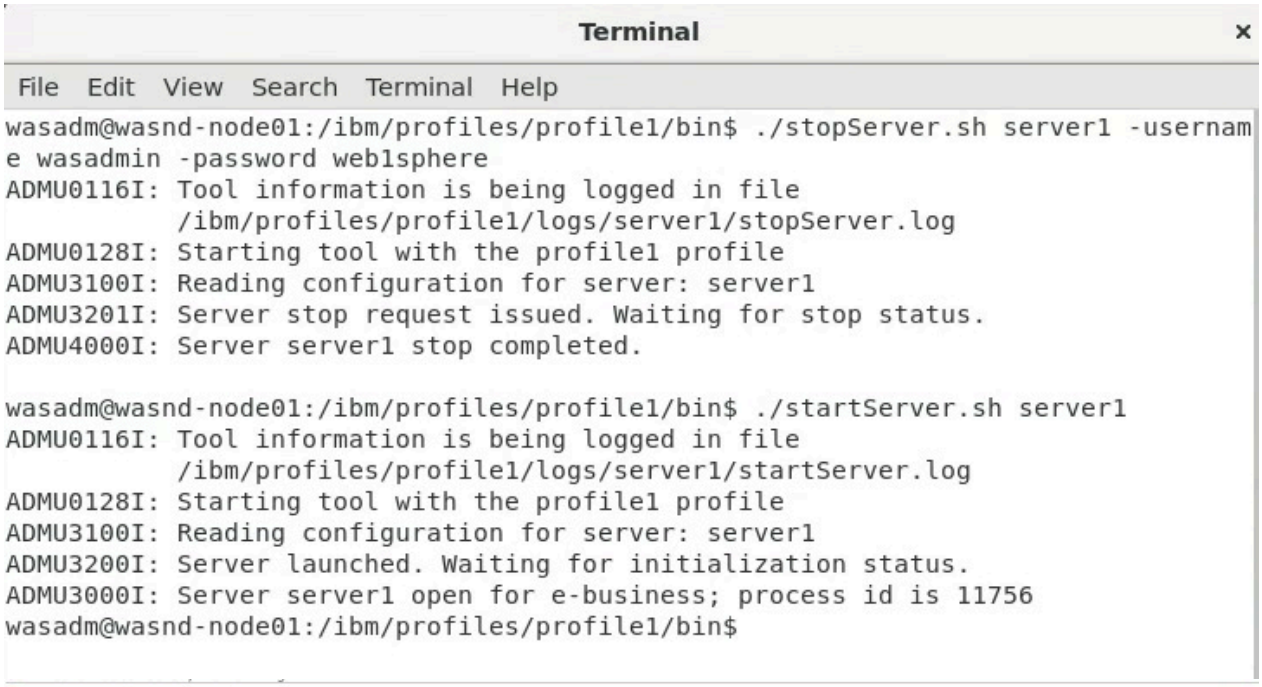
High Performance Extensible Logging (HPEL) is a new mode of logging and tracing. To take advantage of this new log and trace framework, HPEL mode must be enabled. After HPEL mode is enabled, the JVM logs (typically SystemOut.log and SystemErr.log), the trace log (typically trace.log), and the service log (typically activity.log) are no longer written to. Instead, log and trace content is written to a log data or trace data repository in a proprietary binary format and, if configured, to a text log file. By disabling the text log file, you gain the largest possible performance benefit of HPEL. A log viewing tool, Log Viewer, is provided to allow for viewing, filtering, monitoring, and formatting the log and trace data in the repositories.

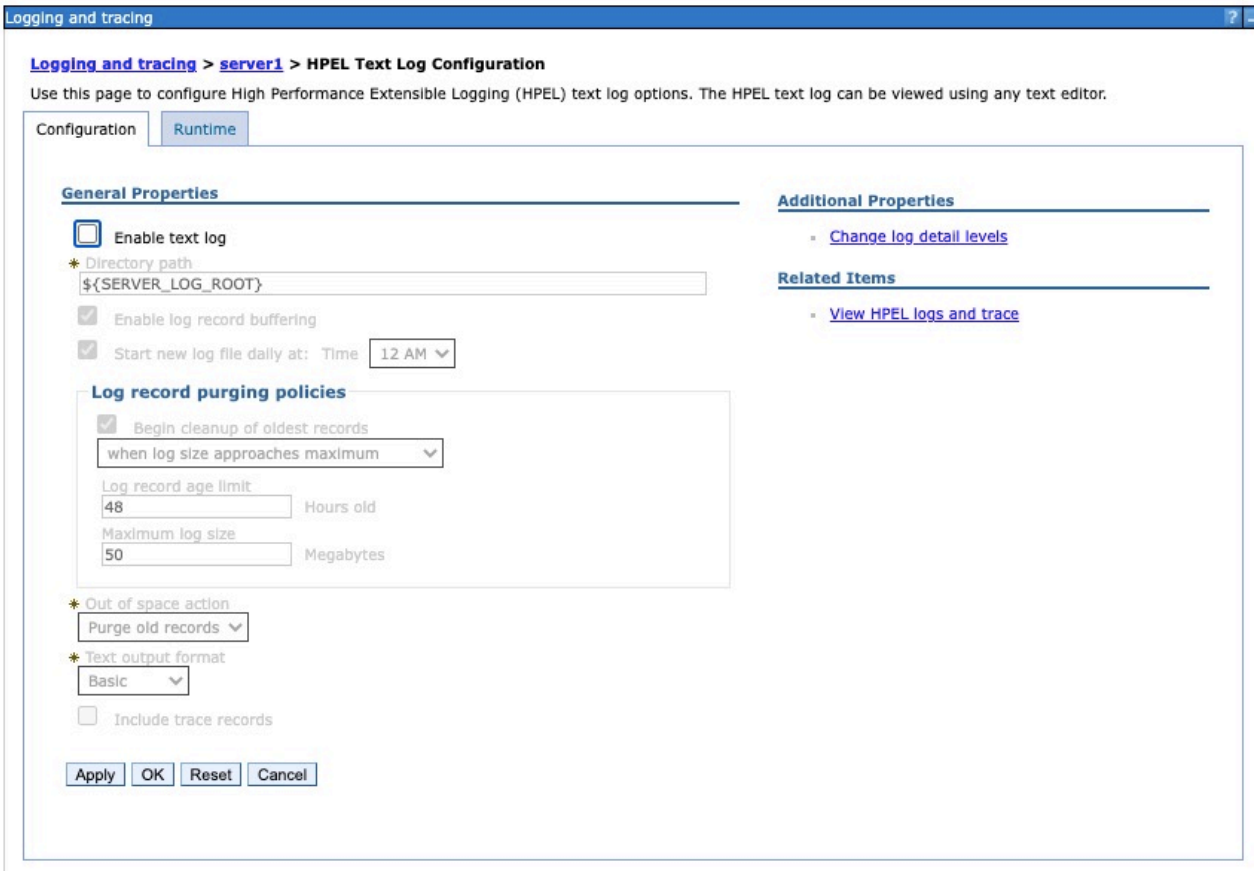
In this section, you enable HPEL mode for server1. Then, you explore and modify the log and trace configurations.

High Performance Extensible Logging (HPEL) is a new mode of logging and tracing. To take advantage of this new log and trace framework, HPEL mode must be enabled. After HPEL mode is enabled, the JVM logs (typically `SystemOut.log` and `SystemErr.log`), the trace log (typically `trace.log`), and the service log (typically `activity.log`) are no longer written to. Instead, log and trace content is written to a log data or trace data repository in a proprietary binary format and, if configured, to a text log file. By disabling the text log file, you gain the largest possible performance benefit of HPEL. A log viewing tool, Log Viewer, is provided to allow for viewing, filtering, monitoring, and formatting the log and trace data in the repositories.

In this section, you enable HPEL mode for server1. Then, you explore and modify the log and trace configurations.




Terminal
File Edit View Search Terminal Help
wasadm@wasnd-node01:/ibm/profiles/profile1/bin$ ./stopServer.sh server1 -usernam
e wasadm -password web1sphere
ADMU0116I: Tool information is being logged in file
/ibm/profiles/profile1/logs/server1/stopServer.log
ADMU0128I: Starting tool with the profile1 profile
ADMU3100I: Reading configuration for server: server1
ADMU3201I: Server stop request issued. Waiting for stop status.
ADMU4000I: Server server1 stop completed.
wasadm@wasnd-node01:/ibm/profiles/profile1/bin$ ./startServer.sh server1
ADMU0116I: Tool information is being logged in file
/ibm/profiles/profile1/logs/server1/startServer.log
ADMU0128I: Starting tool with the profile1 profile
ADMU3100I: Reading configuration for server: server1
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server server1 open for e-business; process id is 11756
wasadm@wasnd-node01:/ibm/profiles/profile1/bin$


Logging and tracing
Logging and tracing > server1 > HPEL Text Log Configuration
Use this page to configure High Performance Extensible Logging (HPEL) text log options. The HPEL text log can be viewed using any text editor.
Configuration
Runtime
General Properties
Enable text log
Directory path
${SERVER_LOG_ROOT}
Enable log record buffering
Start new log file daily at: Time 12 AM
Log record purging policies
Begin cleanup of oldest records
when log size approaches maximum
Log record age limit
48
Hours old
Maximum log size
50
Megabytes
Out of space action
Purge old records
Text output format
Basic
Include trace records
Apply
OK
Reset
Cancel
Additional Properties
Change log detail levels
Related Items
View HPEL logs and trace

Cell=wasnd-node01Cell01, Profile=Dmgr

Logging and tracing

Logging and tracing > **server1** > **HPEL Log Configuration**

Use this page to configure High Performance Extensible Logging (HPEL) log options. The HPEL log can be viewed using the logViewer command (in the profile directory), or using the View HPEL Logs and Trace link.

Configuration | Runtime

General Properties

* Directory path
${SERVER_LOG_ROOT}

☑ Enable log record buffering

☑ Start new log file daily at: Time 12 AM

Log record purging policies

☑ Begin cleanup of oldest records
when log size approaches maximum

Log record age limit
48 Hours old

Maximum log size
20 Megabytes

* Out of space action
Stop logging

Apply | OK | Reset | Cancel

Additional Properties

- Change log detail levels

Related Items

- View HPEL logs and trace

Logging and tracing > **server1** > **HPEL Trace Configuration**

Use this page to configure High Performance Extensible Logging (HPEL) trace options. The HPEL trace can be viewed using the logViewer command (in the profile bin directory), or using the View HPEL Logs and Trace link.

Configuration | Runtime

General Properties

HPEL Trace Output

◉ Trace to a directory

☑ Enable log record buffering

☑ Start new log file daily at: Time 12 AM

Log record purging policies

☑ Begin cleanup of oldest records
when oldest records reach age limit

Log record age limit
12 Hours old

Maximum log size
50 Megabytes

* Out of space action
Purge old records

○ Trace to a memory buffer

* Memory Buffer Size
8 MB

* Directory to use for tracing and dumping memory buffer
${SERVER_LOG_ROOT}

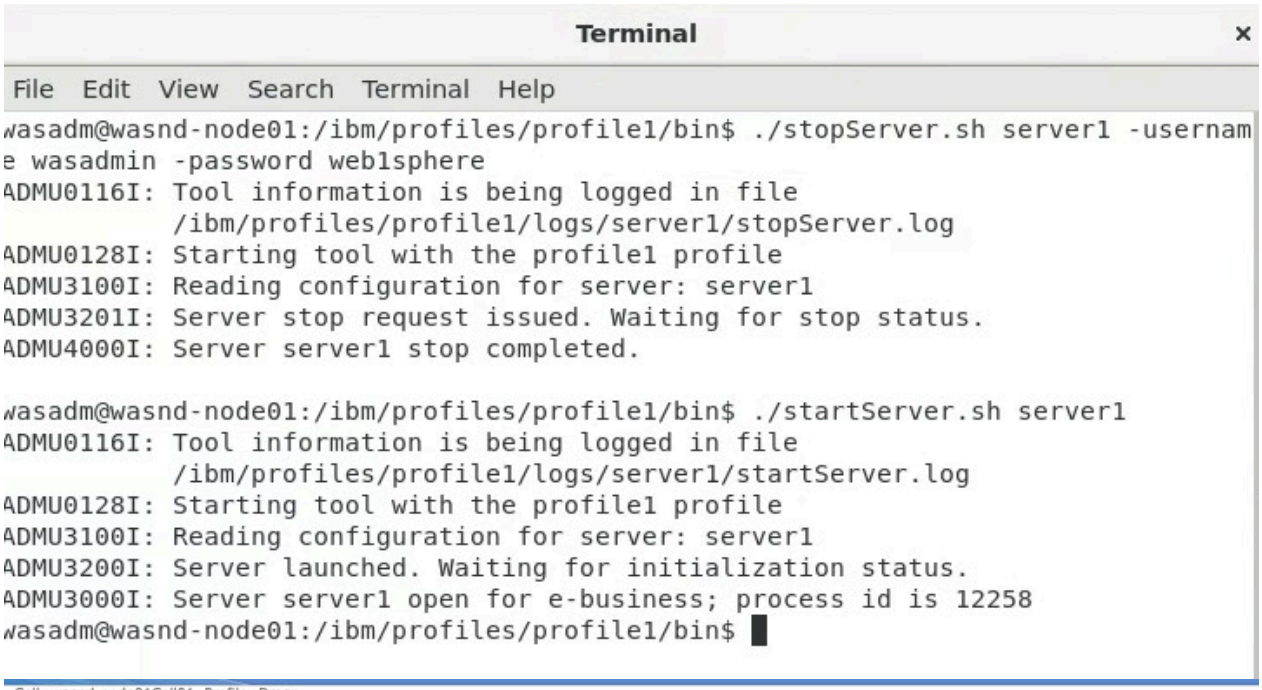
Apply | OK | Reset | Cancel

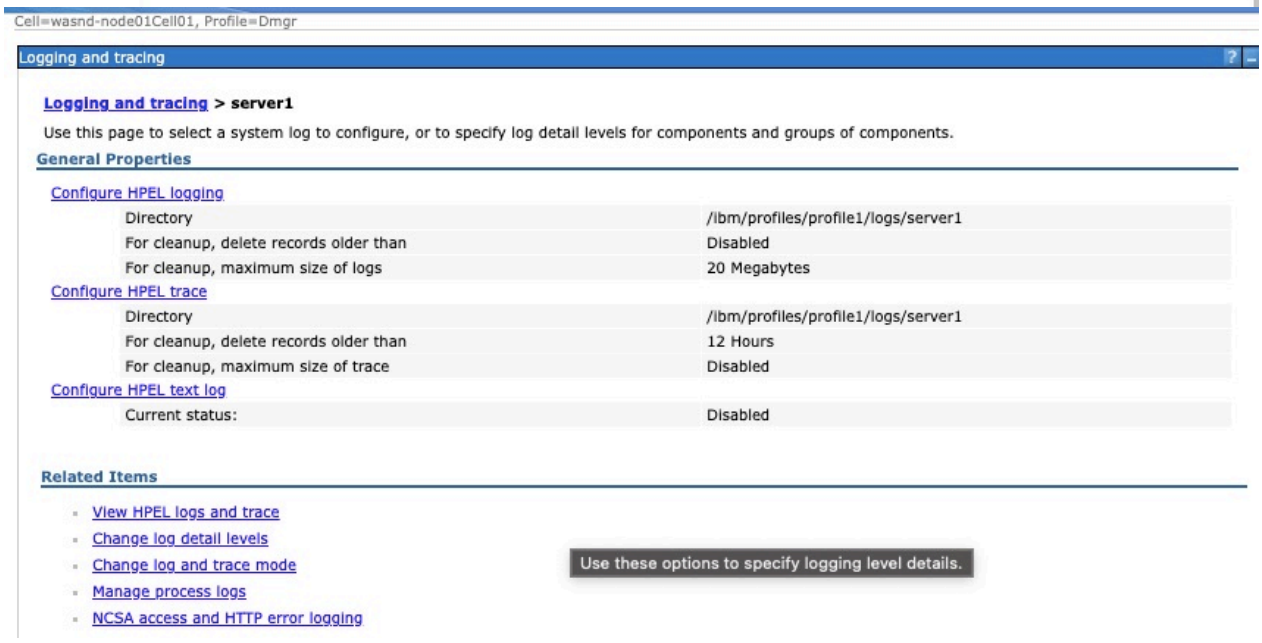
Additional Properties

- Change log detail levels

Related Items

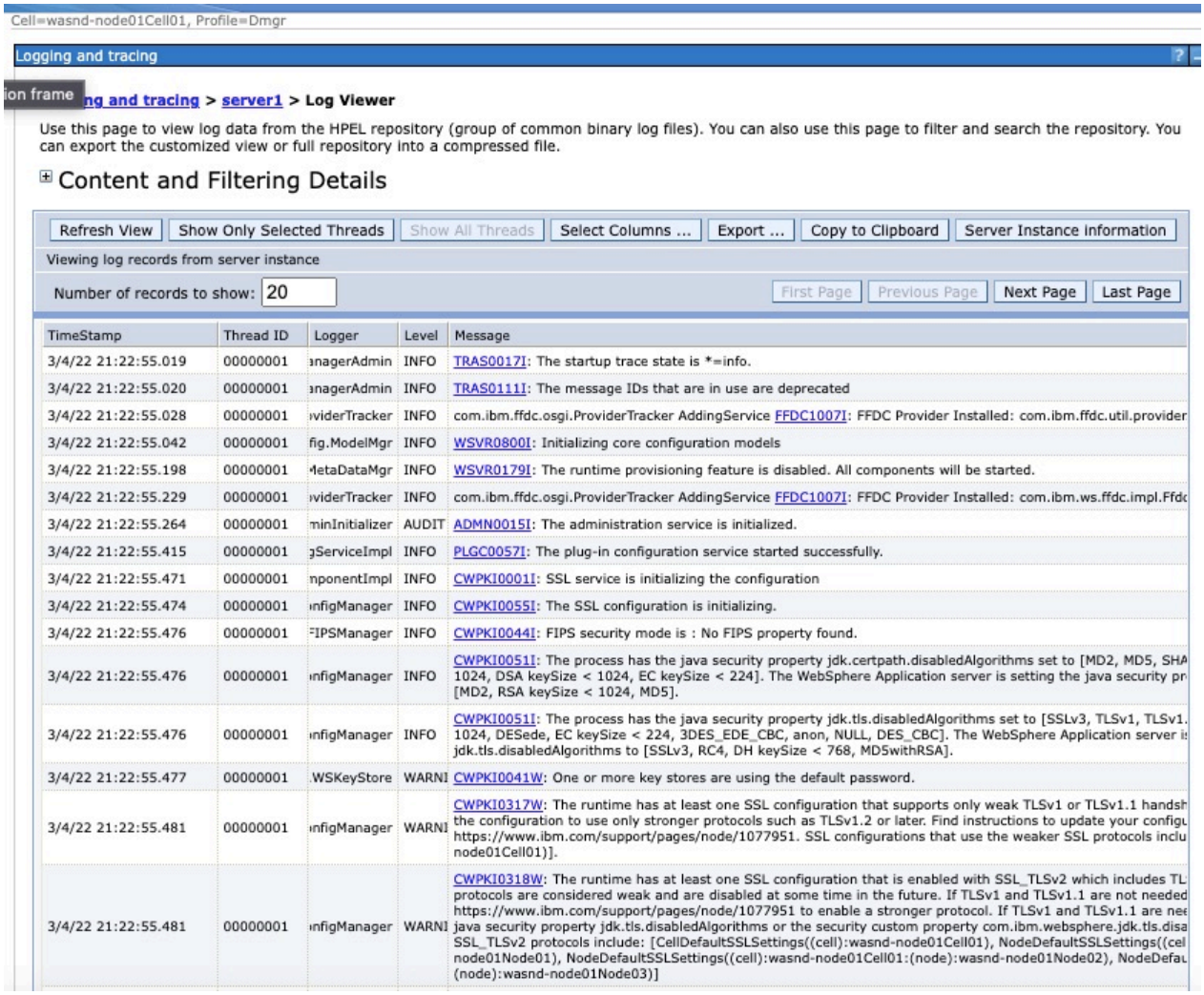
- View HPEL logs and trace





Section 3: Use the Log Viewer in the administrative console to examine log data and trace data

In this section, you use the Log Viewer in the administrative console to examine the log messages for an application server. You use various filtering functions to customize what log records are shown.



Cell=wasnd-node01Cell01, Profile=Dmgr

Logging and tracing

ion frame

ng and tracing > server1 > Log Viewer

Use this page to view log data from the HPEL repository (group of common binary log files). You can also use this page to filter and search the repository. You can export the customized view or full repository into a compressed file.

Content and Filtering Details

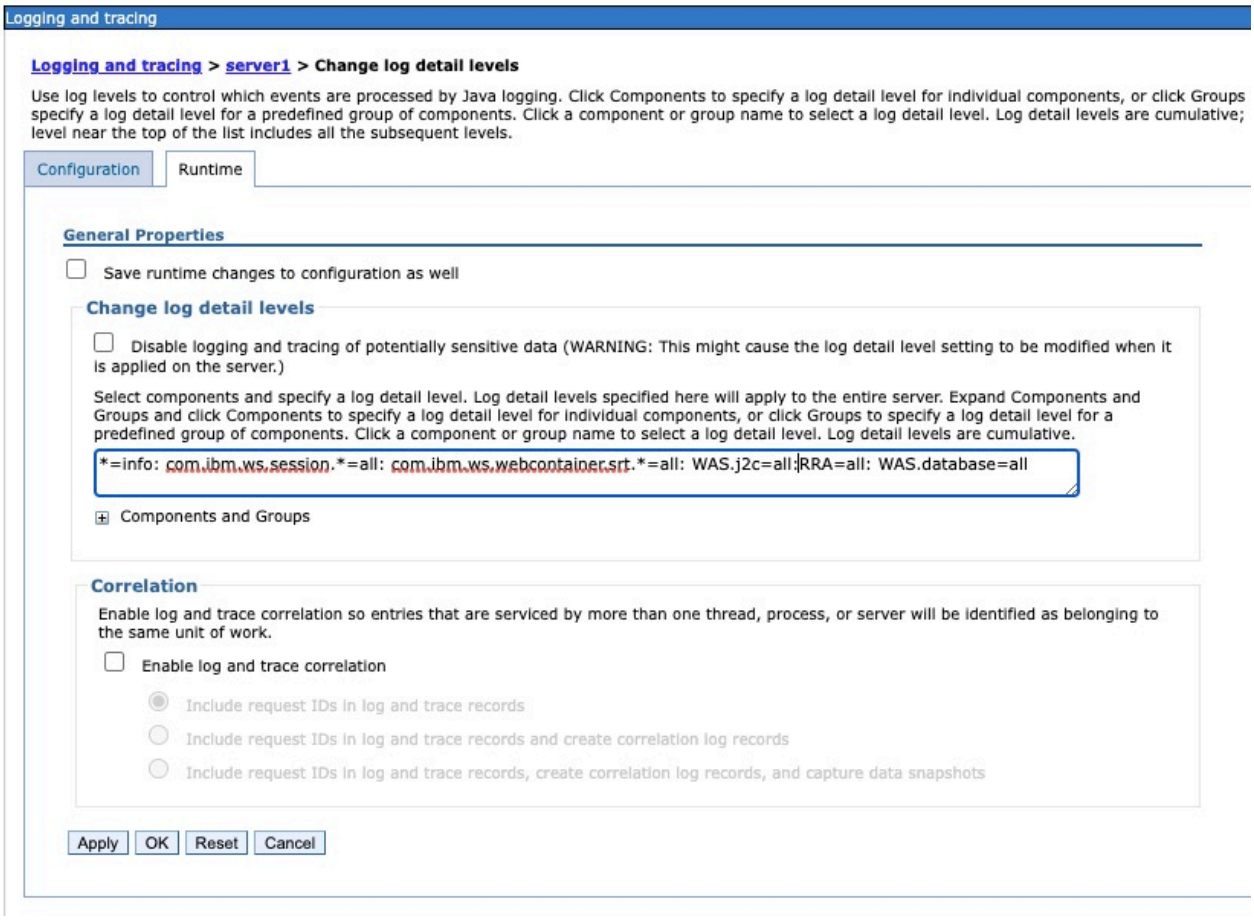
Refresh View | Show Only Selected Threads | Show All Threads | Select Columns ... | Export ... | Copy to Clipboard | Server Instance Information

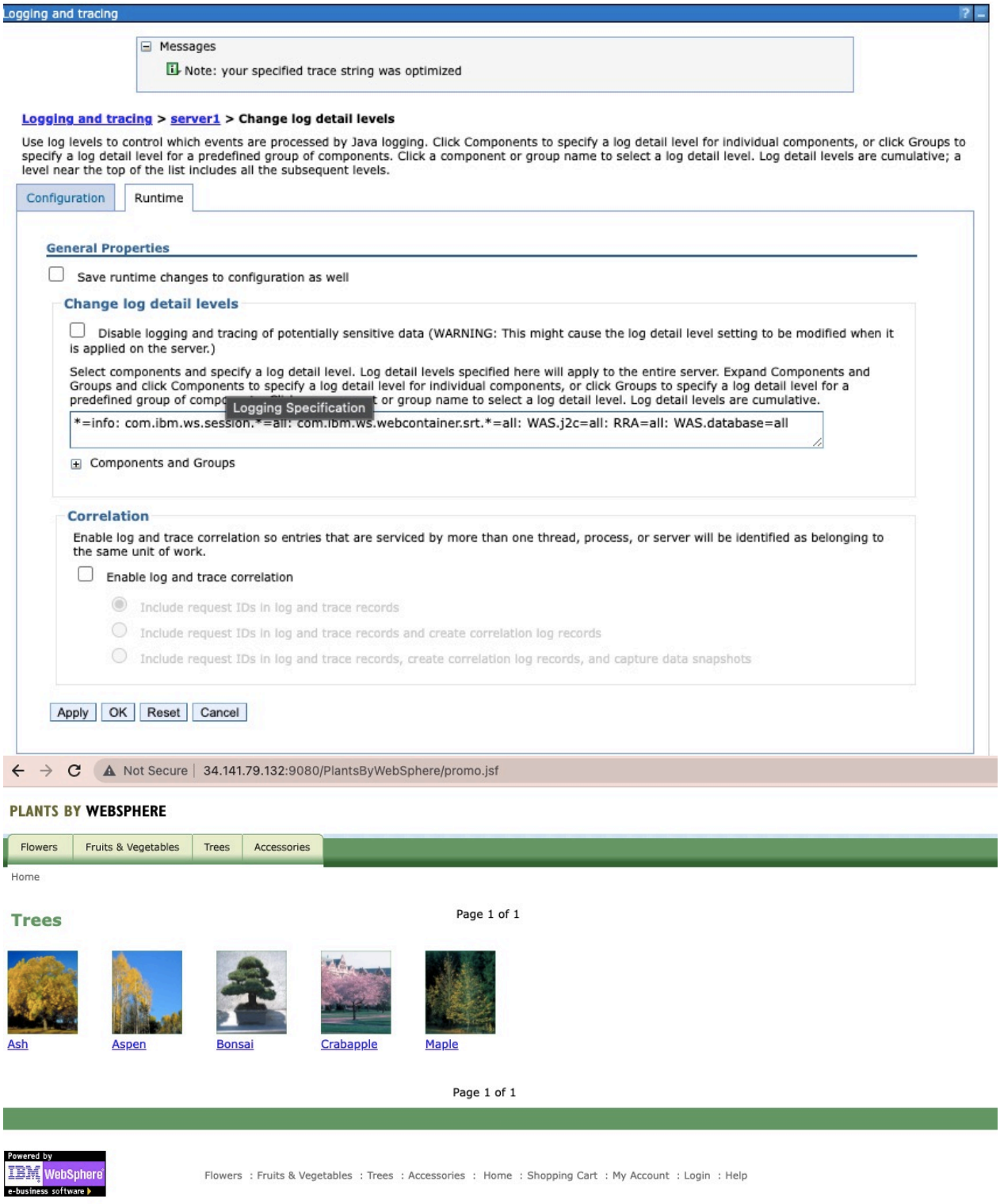
Viewing log records from server instance

Number of records to show: 20

First Page | Previous Page | Next Page | Last Page

TimeStamp	Thread ID	Logger	Level	Message
3/4/22 21:22:55.019	00000001	anagerAdmin	INFO	TRAS0017I: The startup trace state is *=info.
3/4/22 21:22:55.020	00000001	anagerAdmin	INFO	TRAS0111I: The message IDs that are in use are deprecated
3/4/22 21:22:55.028	00000001	viderTracker	INFO	com.ibm.ffdc.osgi.ProviderTracker AddingService FFDC1007I: FFDC Provider Installed: com.ibm.ffdc.util.provider
3/4/22 21:22:55.042	00000001	fig.ModelMgr	INFO	WSVR0800I: Initializing core configuration models
3/4/22 21:22:55.198	00000001	MetaDataMgr	INFO	WSVR0179I: The runtime provisioning feature is disabled. All components will be started.
3/4/22 21:22:55.229	00000001	viderTracker	INFO	com.ibm.ffdc.osgi.ProviderTracker AddingService FFDC1007I: FFDC Provider Installed: com.ibm.ws.ffdc.impl.Ffdc
3/4/22 21:22:55.264	00000001	minInitializer	AUDIT	ADMN0015I: The administration service is initialized.
3/4/22 21:22:55.415	00000001	gServiceImpl	INFO	PLGC0057I: The plug-in configuration service started successfully.
3/4/22 21:22:55.471	00000001	nponentImpl	INFO	CWPKI0001I: SSL service is initializing the configuration
3/4/22 21:22:55.474	00000001	nfigManager	INFO	CWPKI0055I: The SSL configuration is initializing.
3/4/22 21:22:55.476	00000001	FIPSManager	INFO	CWPKI0044I: FIPS security mode is : No FIPS property found.
3/4/22 21:22:55.476	00000001	nfigManager	INFO	CWPKI0051I: The process has the java security property jdk.certpath.disabledAlgorithms set to [MD2, MD5, SHA 1024, DSA keySize < 1024, EC keySize < 224]. The WebSphere Application server is setting the java security pr [MD2, RSA keySize < 1024, MD5].
3/4/22 21:22:55.476	00000001	nfigManager	INFO	CWPKI0051I: The process has the java security property jdk.tls.disabledAlgorithms set to [SSLv3, TLSv1, TLSv1. 1024, DESede, EC keySize < 224, 3DES_EDE_CBC, anon, NULL, DES_CBC]. The WebSphere Application server is jdk.tls.disabledAlgorithms to [SSLv3, RC4, DH keySize < 768, MD5withRSA].
3/4/22 21:22:55.477	00000001	.WSKeyStore	WARNI	CWPKI0041W: One or more key stores are using the default password.
3/4/22 21:22:55.481	00000001	nfigManager	WARNI	CWPKI0317W: The runtime has at least one SSL configuration that supports only weak TLSv1 or TLSv1.1 handsh the configuration to use only stronger protocols such as TLSv1.2 or later. Find instructions to update your configu https://www.ibm.com/support/pages/node/1077951. SSL configurations that use the weaker SSL protocols inclu node01Cell01)].
3/4/22 21:22:55.481	00000001	nfigManager	WARNI	CWPKI0318W: The runtime has at least one SSL configuration that is enabled with SSL_TLSv2 which includes TL protocols are considered weak and are disabled at some time in the future. If TLSv1 and TLSv1.1 are not needed https://www.ibm.com/support/pages/node/1077951 to enable a stronger protocol. If TLSv1 and TLSv1.1 are nee java security property jdk.tls.disabledAlgorithms or the security custom property com.ibm.websphere.jdk.tls.disa SSL_TLSv2 protocols include: [CellDefaultSSLSettings((cell):wasnd-node01Cell01), NodeDefaultSSLSettings((cel node01Node01), NodeDefaultSSLSettings((cell):wasnd-node01Cell01:(node):wasnd-node01Node02), NodeDefau (node):wasnd-node01Node03)]


Logging and tracing
Logging and tracing > server1 > Change log detail levels
Use log levels to control which events are processed by Java logging. Click Components to specify a log detail level for individual components, or click Groups specify a log detail level for a predefined group of components. Click a component or group name to select a log detail level. Log detail levels are cumulative; level near the top of the list includes all the subsequent levels.
Configuration
Runtime
General Properties
Save runtime changes to configuration as well
Change log detail levels
Disable logging and tracing of potentially sensitive data (WARNING: This might cause the log detail level setting to be modified when it is applied on the server.)
Select components and specify a log detail level. Log detail levels specified here will apply to the entire server. Expand Components and Groups and click Components to specify a log detail level for individual components, or click Groups to specify a log detail level for a predefined group of components. Click a component or group name to select a log detail level. Log detail levels are cumulative.
=info: com.ibm.ws.session.=all: com.ibm.ws.webcontainer.srt.*=all: WAS.j2c=all:RRA=all: WAS.database=all
Components and Groups
Correlation
Enable log and trace correlation so entries that are serviced by more than one thread, process, or server will be identified as belonging to the same unit of work.
Enable log and trace correlation
Include request IDs in log and trace records
Include request IDs in log and trace records and create correlation log records
Include request IDs in log and trace records, create correlation log records, and capture data snapshots
Apply
OK
Reset
Cancel



Section 4: Enable tracing for an application server and view trace data from the Log Viewer

In this section, you configure tracing on the session management components of server1. Use the PlantsByWebSphere application to generate trace data, and view the trace data in the Log Viewer.

Section 5: Enable cross-component trace (XCT)

In this section, you learn how to enable cross-component trace (XCT) for an application server. You also examine the request IDs and other data that XCT provides in the server logs.

Include request IDs and trace records and create correlation log records

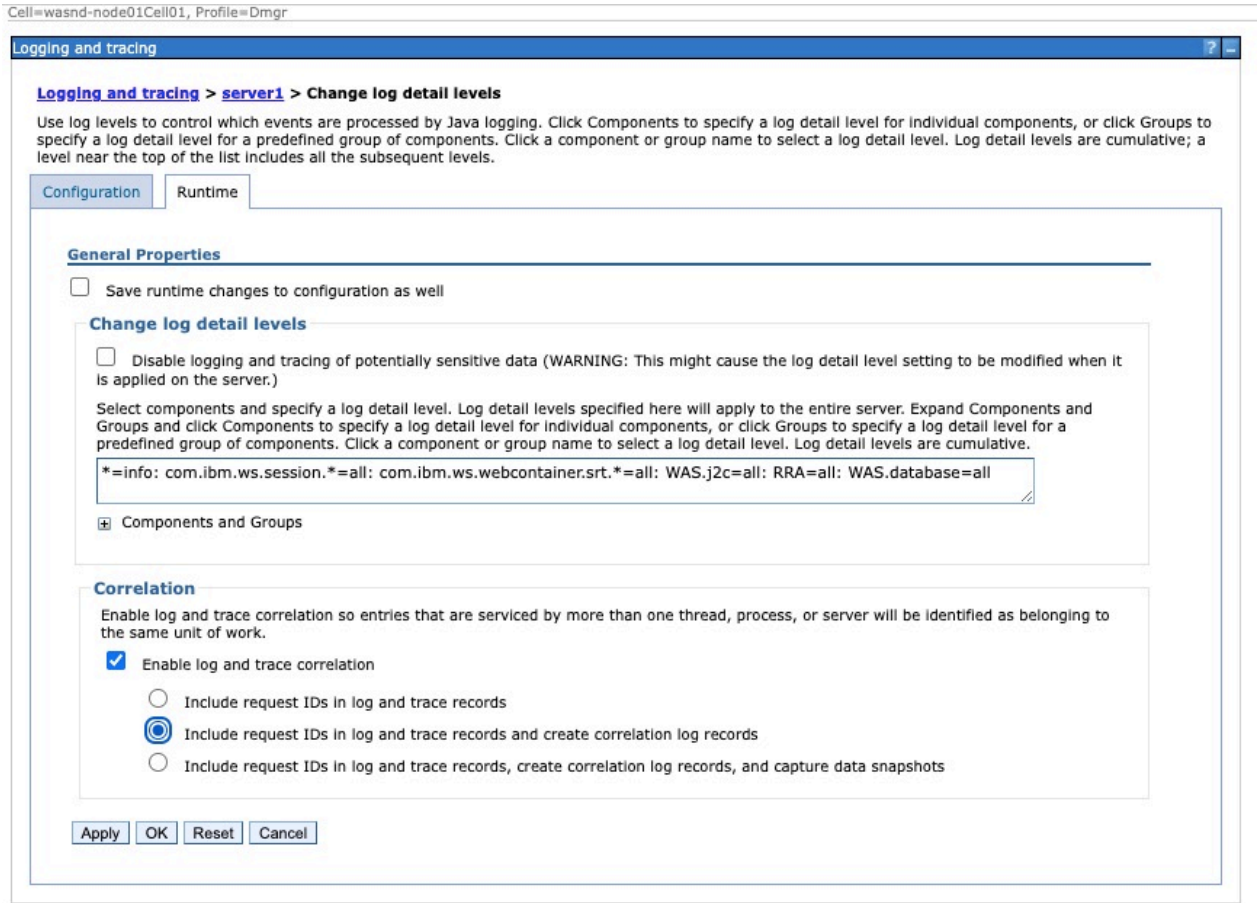
This setting enables XCT to include request IDs in log and trace files when you want to see which log and trace entries, in all threads and application server processes, are related to the same request. Request IDs are recorded only when using HPEL log and trace mode and can be seen or used for filtering when using the logViewer command.

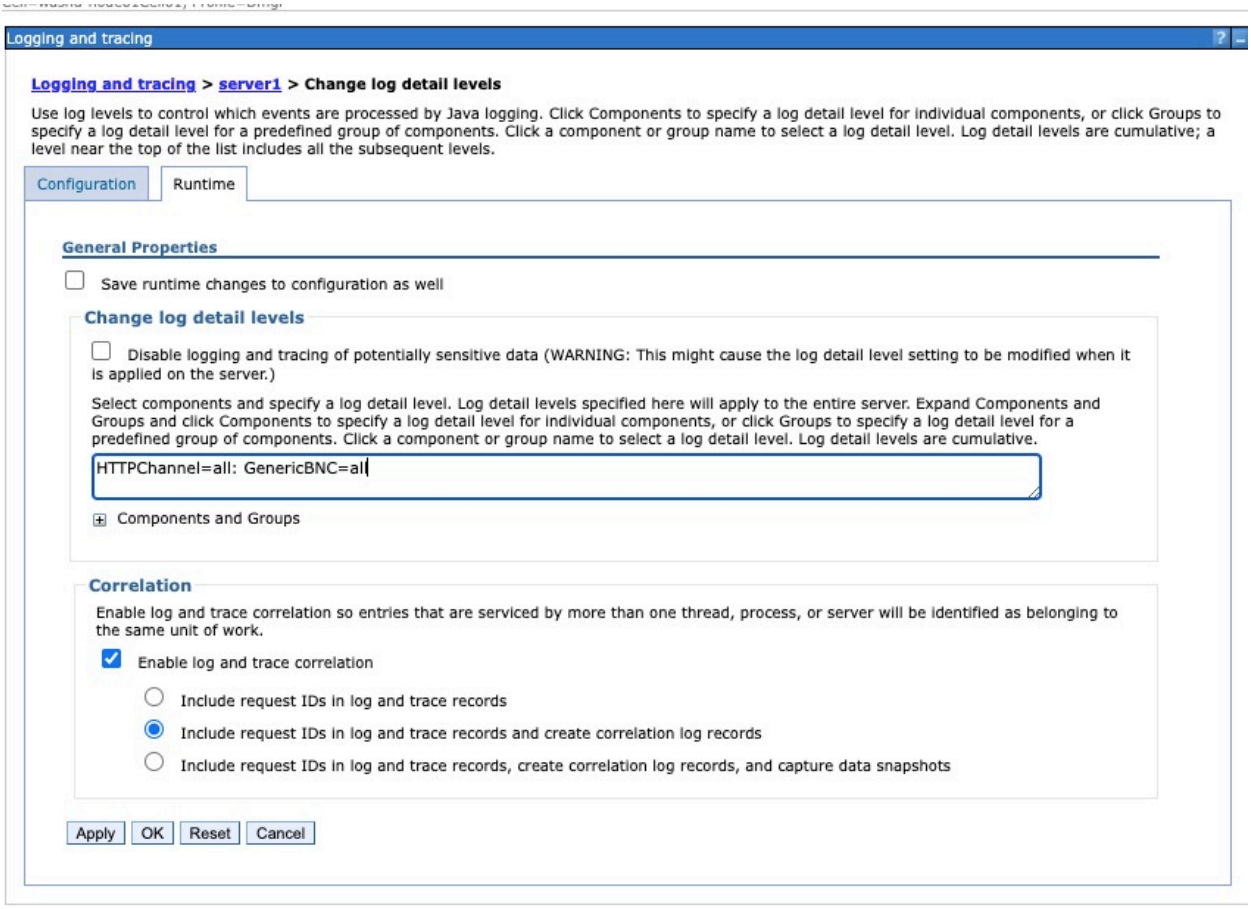
In addition, XCT creates correlation log records when you want to log how requests branch between threads and processes, and see extra information about each request.

Warning: Enabling XCT to create correlation log records might have a significant performance cost on your system, so is best suited to test and development environments.

IBM Cross Component Trace Log Viewer

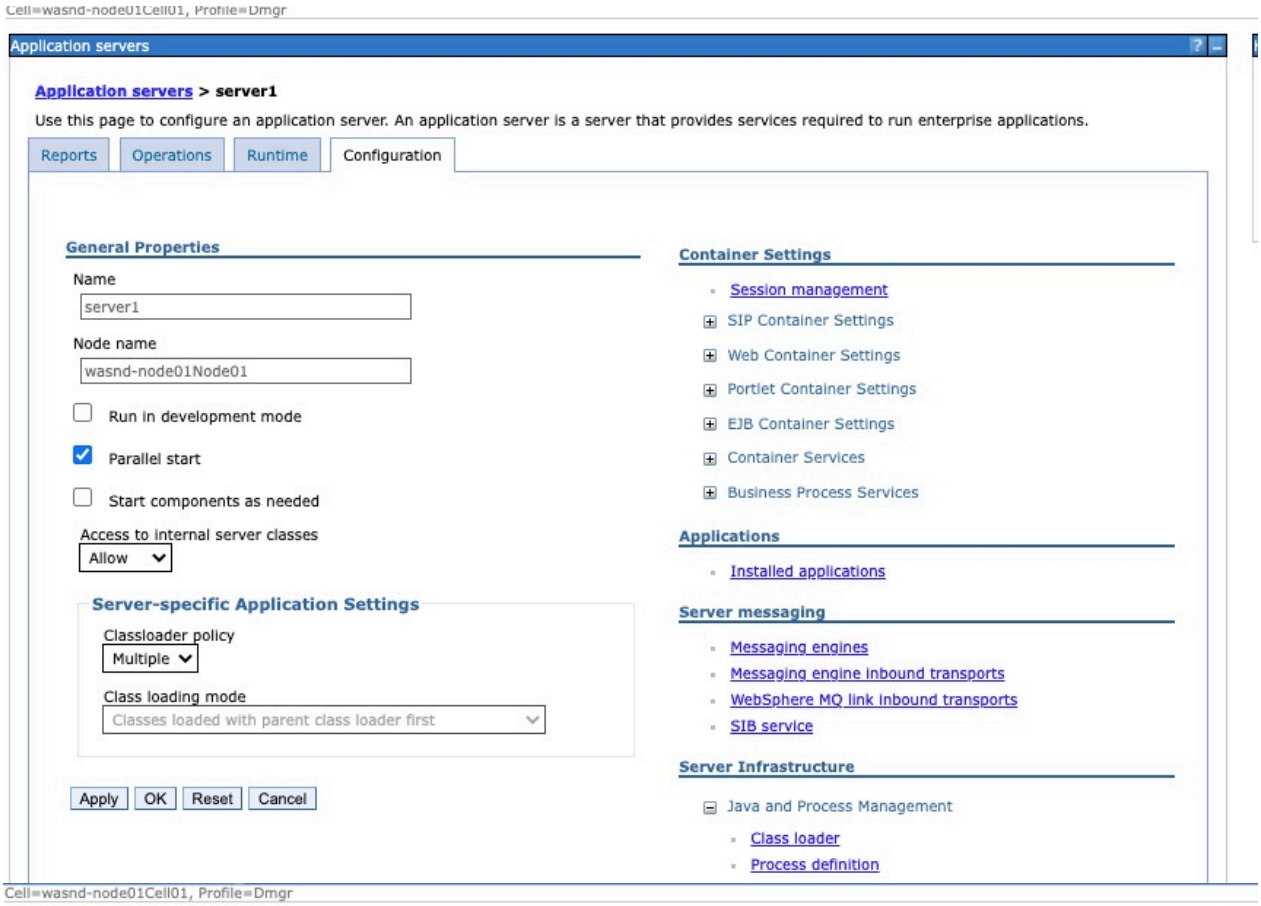
Available in the IBM Support Assistant, IBM WebSphere Cross Component Trace Log Viewer provides enhanced log file views for logs that are augmented with Cross Component Trace correlation log records. Logs can be displayed in flat or hierarchical layouts, and multiple logs can be loaded and viewed simultaneously with log entries related to each request conveniently grouped.

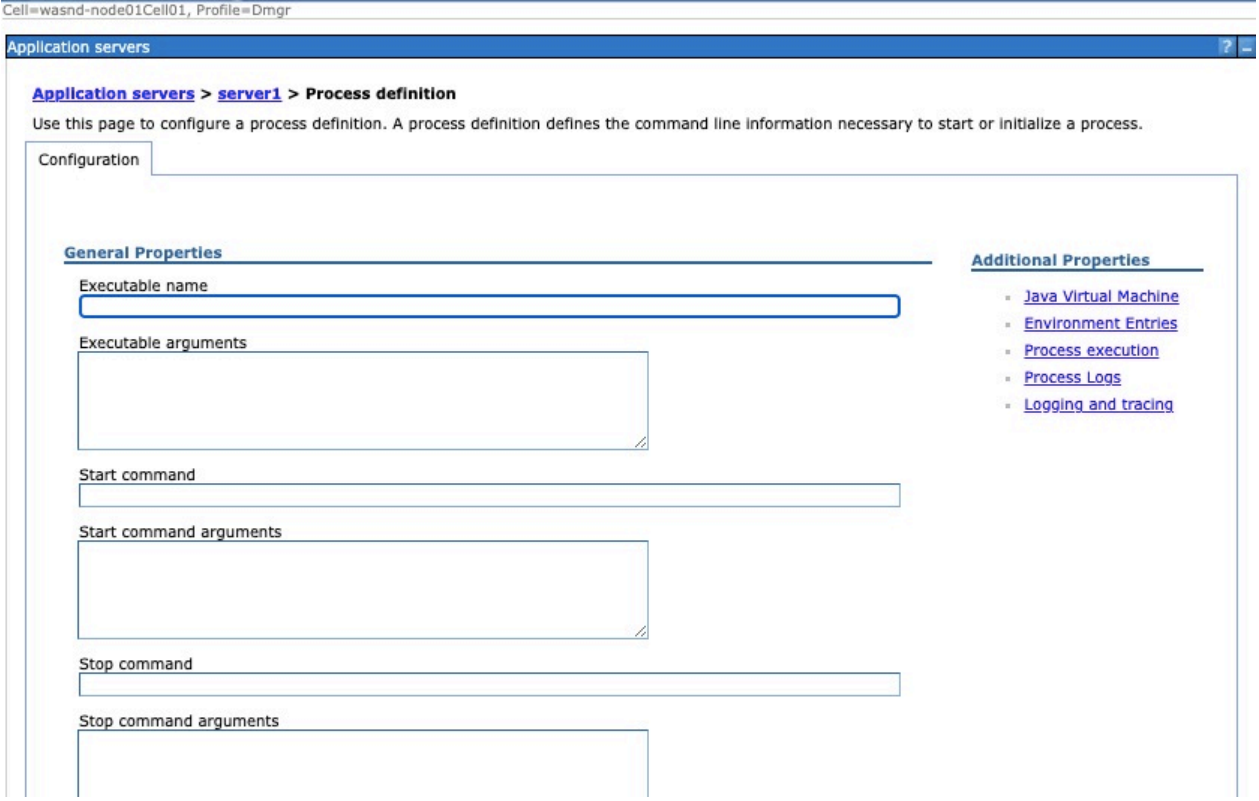

Cell=wasnd-node01Cell01, Profile=Dmgr
Logging and tracing
Logging and tracing > server1 > Change log detail levels
Use log levels to control which events are processed by Java logging. Click Components to specify a log detail level for individual components, or click Groups to specify a log detail level for a predefined group of components. Click a component or group name to select a log detail level. Log detail levels are cumulative; a level near the top of the list includes all the subsequent levels.
Configuration
Runtime
General Properties
Save runtime changes to configuration as well
Change log detail levels
Disable logging and tracing of potentially sensitive data (WARNING: This might cause the log detail level setting to be modified when it is applied on the server.)
Select components and specify a log detail level. Log detail levels specified here will apply to the entire server. Expand Components and Groups and click Components to specify a log detail level for individual components, or click Groups to specify a log detail level for a predefined group of components. Click a component or group name to select a log detail level. Log detail levels are cumulative.
=info: com.ibm.ws.session.=all: com.ibm.ws.webcontainer.srt.*=all: WAS.j2c=all: RRA=all: WAS.database=all
Components and Groups
Correlation
Enable log and trace correlation so entries that are serviced by more than one thread, process, or server will be identified as belonging to the same unit of work.
Enable log and trace correlation
Include request IDs in log and trace records
Include request IDs in log and trace records and create correlation log records
Include request IDs in log and trace records, create correlation log records, and capture data snapshots
Apply
OK
Reset
Cancel

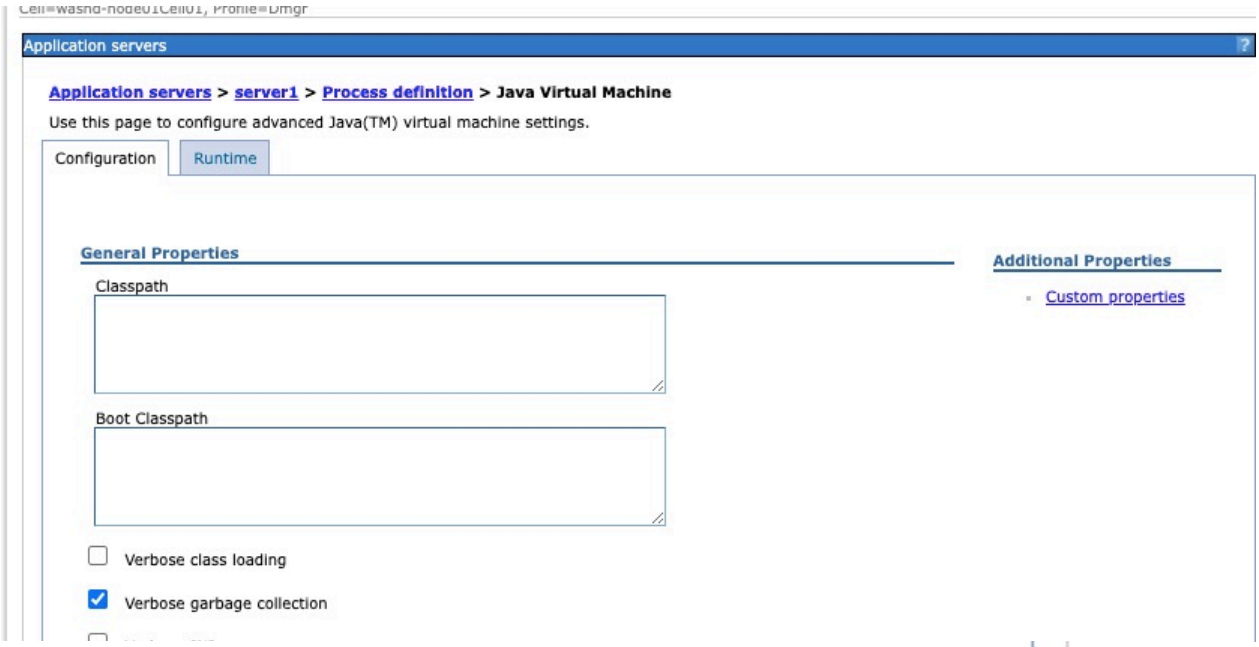


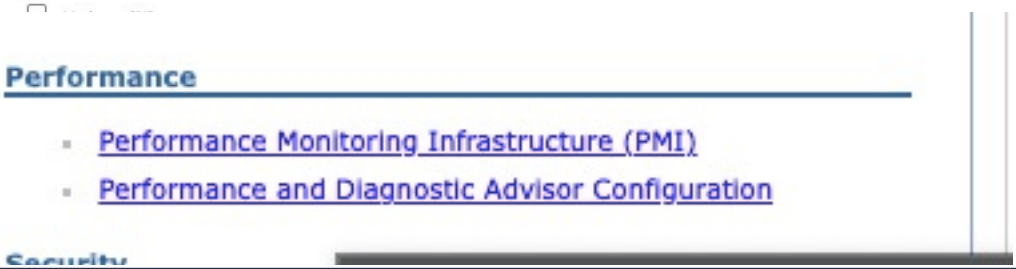
Section 6: Collecting JVM data

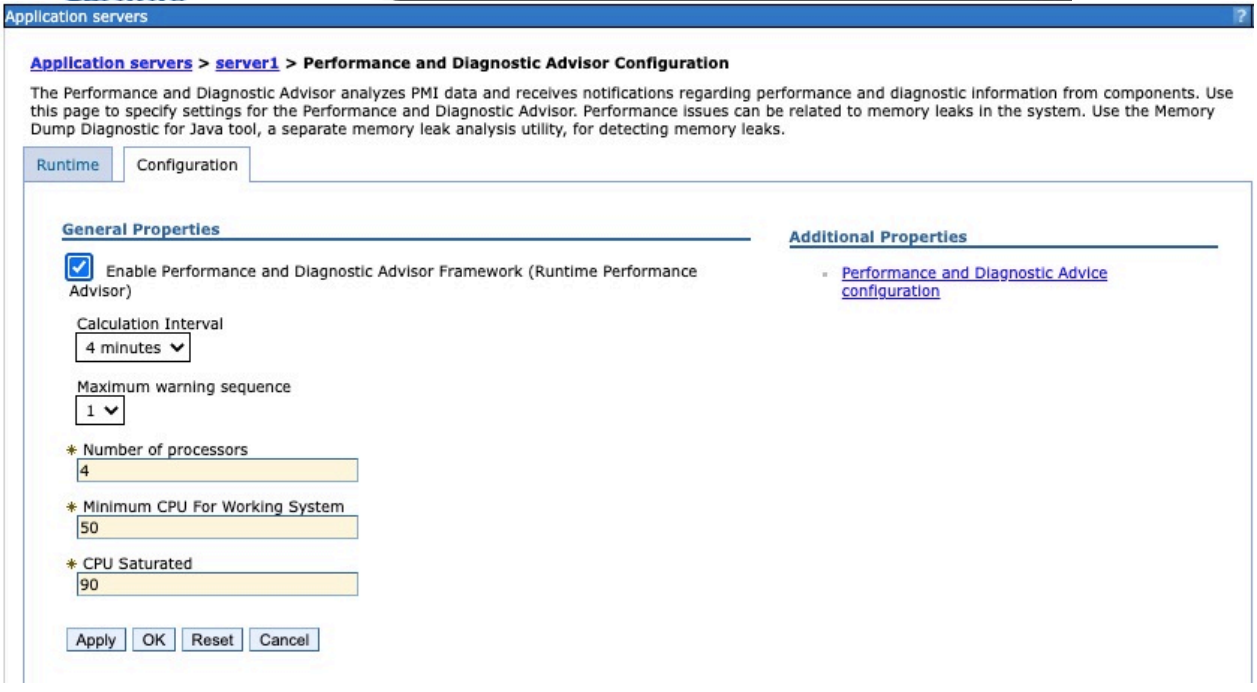
There are several common JVM-related problems such as hung threads, memory leaks, and out-of-memory conditions. This section shows you how to collect diagnostic data to help troubleshoot these problems. First, you install an example application that is written to illustrate several JVM-related problems.

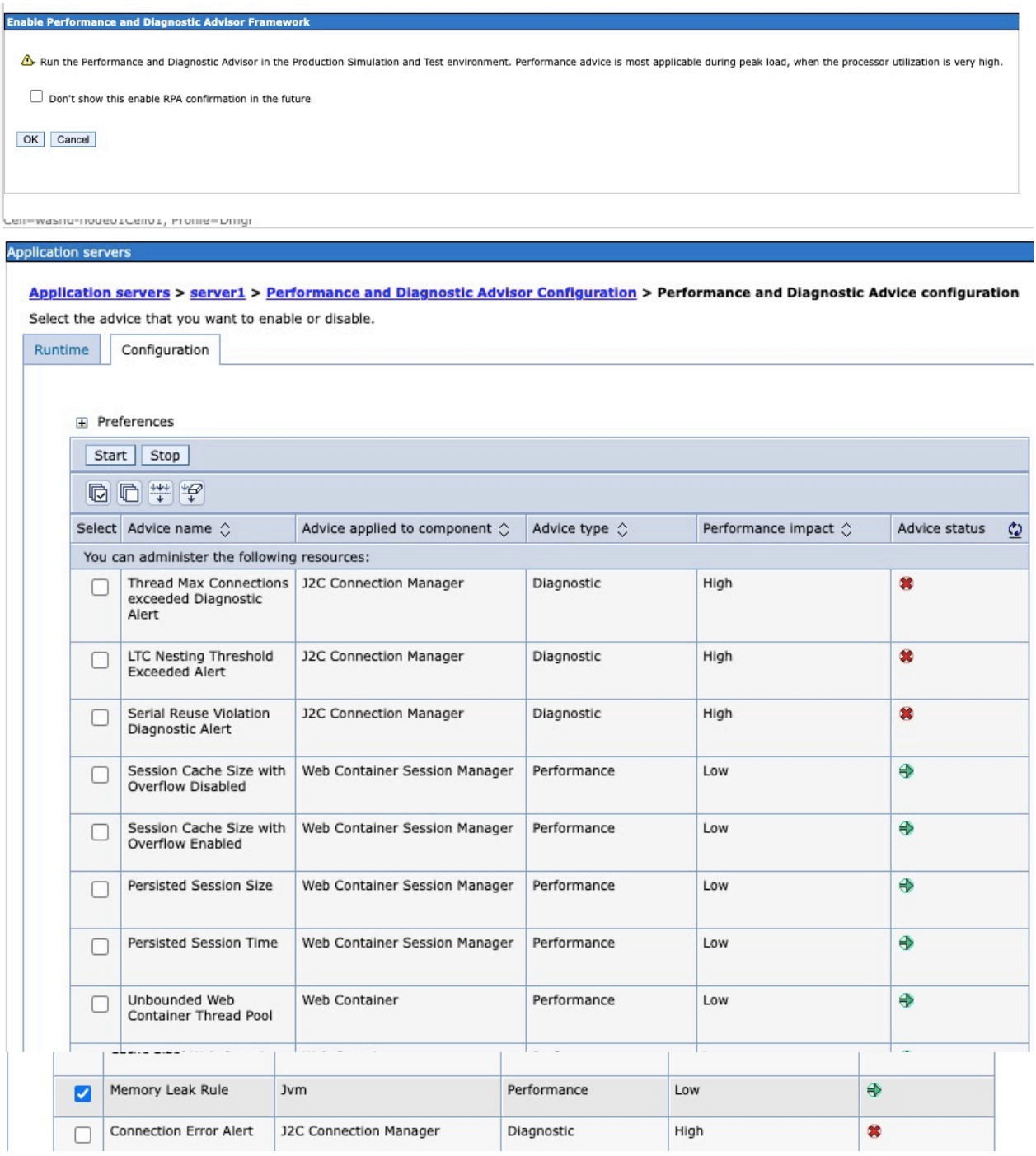

Cell=wasnd-node01Cell01, Profile=Dmgr
Application servers
Application servers > server1
Use this page to configure an application server. An application server is a server that provides services required to run enterprise applications.
Reports
Operations
Runtime
Configuration
General Properties
Name
server1
Node name
wasnd-node01Node01
Run in development mode
Parallel start
Start components as needed
Access to internal server classes
Allow
Server-specific Application Settings
Classloader policy
Multiple
Class loading mode
Classes loaded with parent class loader first
Apply
OK
Reset
Cancel
Container Settings
Session management
SIP Container Settings
Web Container Settings
Portlet Container Settings
EJB Container Settings
Container Services
Business Process Services
Applications
Installed applications
Server messaging
Messaging engines
Messaging engine inbound transports
WebSphere MQ link inbound transports
SIB service
Server Infrastructure
Java and Process Management
Class loader
Process definition

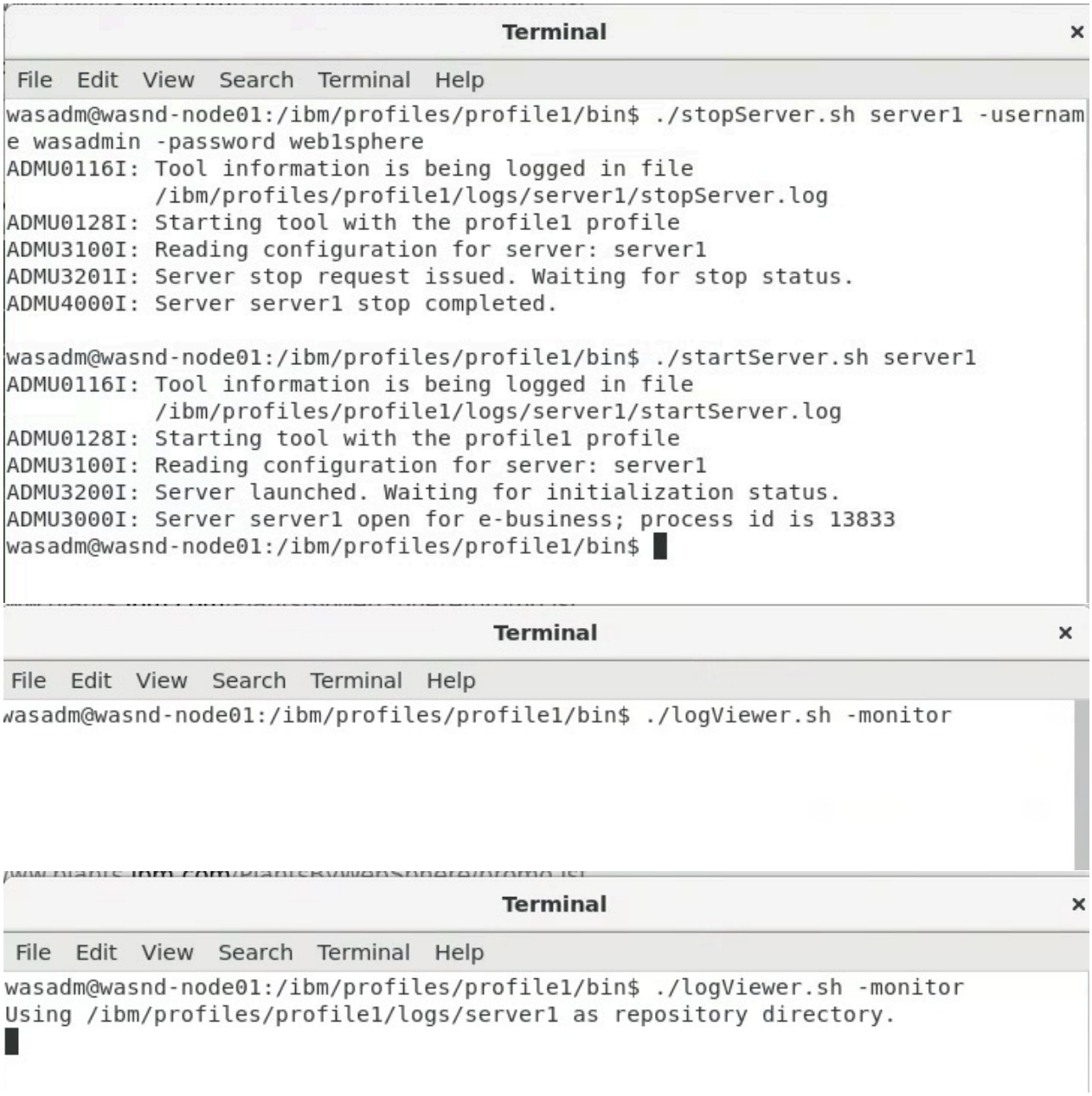

Cell=wasnd-node01Cell01, Profile=Dmgr
Application servers
Application servers > server1 > Process definition
Use this page to configure a process definition. A process definition defines the command line information necessary to start or initialize a process.
Configuration
General Properties
Executable name
Executable arguments
Start command
Start command arguments
Stop command
Stop command arguments
Additional Properties
Java Virtual Machine
Environment Entries
Process execution
Process Logs
Logging and tracing


Application servers
Application servers > server1 > Process definition > Java Virtual Machine
Use this page to configure advanced Java(TM) virtual machine settings.
Configuration
Runtime
General Properties
Additional Properties
Custom properties
Classpath
Boot Classpath
Verbose class loading
Verbose garbage collection


Performance
Performance Monitoring Infrastructure (PMI)
Performance and Diagnostic Advisor Configuration

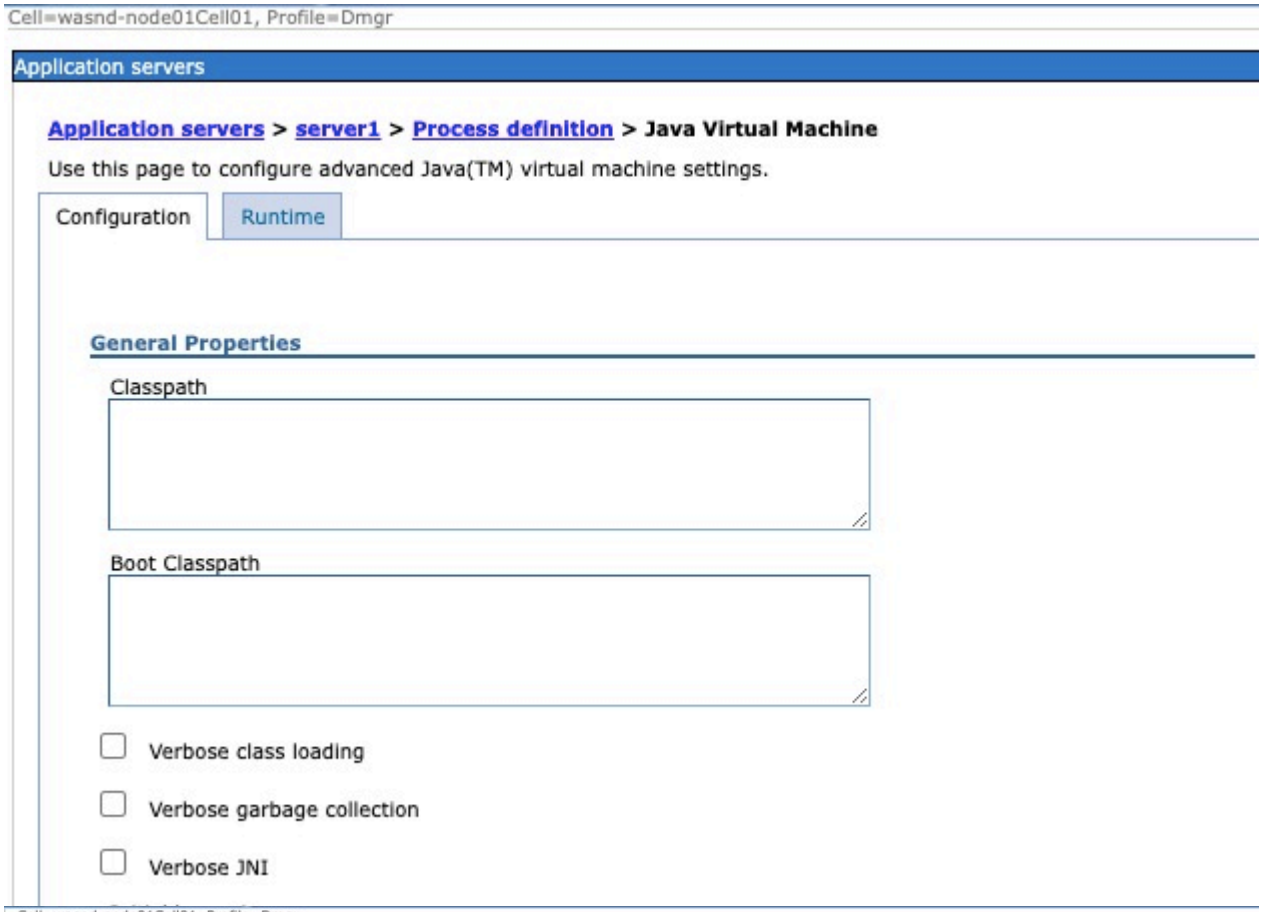

Application servers
Application servers > server1 > Performance and Diagnostic Advisor Configuration
The Performance and Diagnostic Advisor analyzes PMI data and receives notifications regarding performance and diagnostic information from components. Use this page to specify settings for the Performance and Diagnostic Advisor. Performance issues can be related to memory leaks in the system. Use the Memory Dump Diagnostic for Java tool, a separate memory leak analysis utility, for detecting memory leaks.
Runtime
Configuration
General Properties
Additional Properties
Performance and Diagnostic Advice configuration
Enable Performance and Diagnostic Advisor Framework (Runtime Performance Advisor)
Calculation Interval
4 minutes
Maximum warning sequence
1
* Number of processors
4
* Minimum CPU For Working System
50
* CPU Saturated
90
Apply
OK
Reset
Cancel

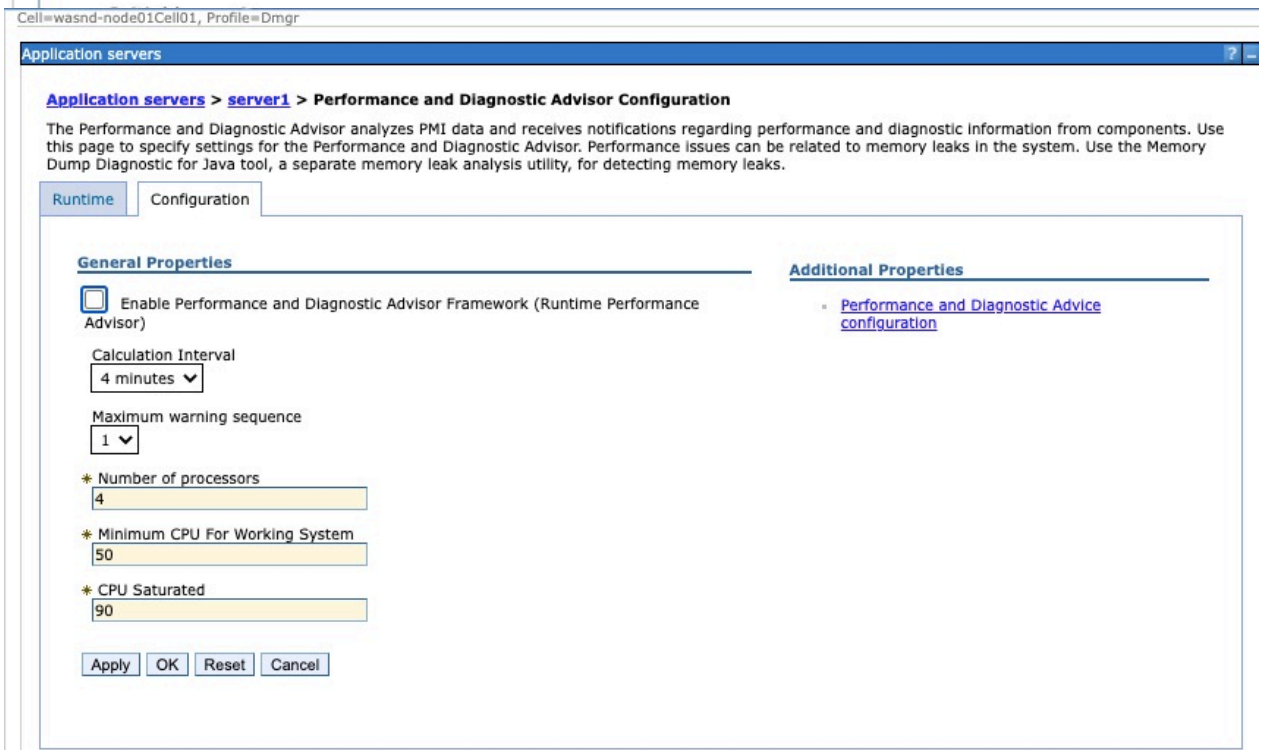

Enable Performance and Diagnostic Advisor Framework
Run the Performance and Diagnostic Advisor in the Production Simulation and Test environment. Performance advice is most applicable during peak load, when the processor utilization is very high.
Don't show this enable RPA confirmation in the future
OK
Cancel
Application servers
Application servers > server1 > Performance and Diagnostic Advisor Configuration > Performance and Diagnostic Advice configuration
Select the advice that you want to enable or disable.
Runtime
Configuration
Preferences
Start
Stop
Select
Advice name
Advice applied to component
Advice type
Performance impact
Advice status
You can administer the following resources:
Thread Max Connections exceeded Diagnostic Alert
J2C Connection Manager
Diagnostic
High
LTC Nesting Threshold Exceeded Alert
J2C Connection Manager
Diagnostic
High
Serial Reuse Violation Diagnostic Alert
J2C Connection Manager
Diagnostic
High
Session Cache Size with Overflow Disabled
Web Container Session Manager
Performance
Low
Session Cache Size with Overflow Enabled
Web Container Session Manager
Performance
Low
Persisted Session Size
Web Container Session Manager
Performance
Low
Persisted Session Time
Web Container Session Manager
Performance
Low
Unbounded Web Container Thread Pool
Web Container
Performance
Low
Memory Leak Rule
Jvm
Performance
Low
Connection Error Alert
J2C Connection Manager
Diagnostic
High



Section 7: Clean up server1

The last section concludes the active exercise. The next section is read-only. Follow these steps to clean up server1 and uninstall the BadApp application.


Cell=wasnd-node01Cell01, Profile=Dmgr
Application servers
Application servers > server1 > Process definition > Java Virtual Machine
Use this page to configure advanced Java(TM) virtual machine settings.
Configuration
Runtime
General Properties
Classpath
Boot Classpath
Verbose class loading
Verbose garbage collection
Verbose JNI


Cell=wasnd-node01Cell01, Profile=Dmgr
Application servers
Application servers > server1 > Performance and Diagnostic Advisor Configuration
The Performance and Diagnostic Advisor analyzes PMI data and receives notifications regarding performance and diagnostic information from components. Use this page to specify settings for the Performance and Diagnostic Advisor. Performance issues can be related to memory leaks in the system. Use the Memory Dump Diagnostic for Java tool, a separate memory leak analysis utility, for detecting memory leaks.
Runtime
Configuration
General Properties
Enable Performance and Diagnostic Advisor Framework (Runtime Performance Advisor)
Calculation Interval
4 minutes
Maximum warning sequence
1
Number of processors
4
Minimum CPU For Working System
50
CPU Saturated
90
Apply
OK
Reset
Cancel
Additional Properties
Performance and Diagnostic Advice configuration