Module 4

Managing and Monitoring a WebLogic Server Environment

At the end of this module you will be able to:

- ✓ Understand machines and Node Manager
- ✓ Describe Simple Logging
- ✓ Use commands to get attributes from an MBean
- ✓ Explain basic SNMP concepts
- ✓ Configure the WLS SNMP agent
- ✓ Use the WLS SNMP management command-line tools

Road Map

1. **Remote Administration**
   - Configuring Machines
   - Node Manager
   - Configuring Node Manager
2. Logs and Monitoring
3. SNMP Concepts
4. WLS SNMP Agent
5. WLS SNMP Management Tools
6. Network Channels

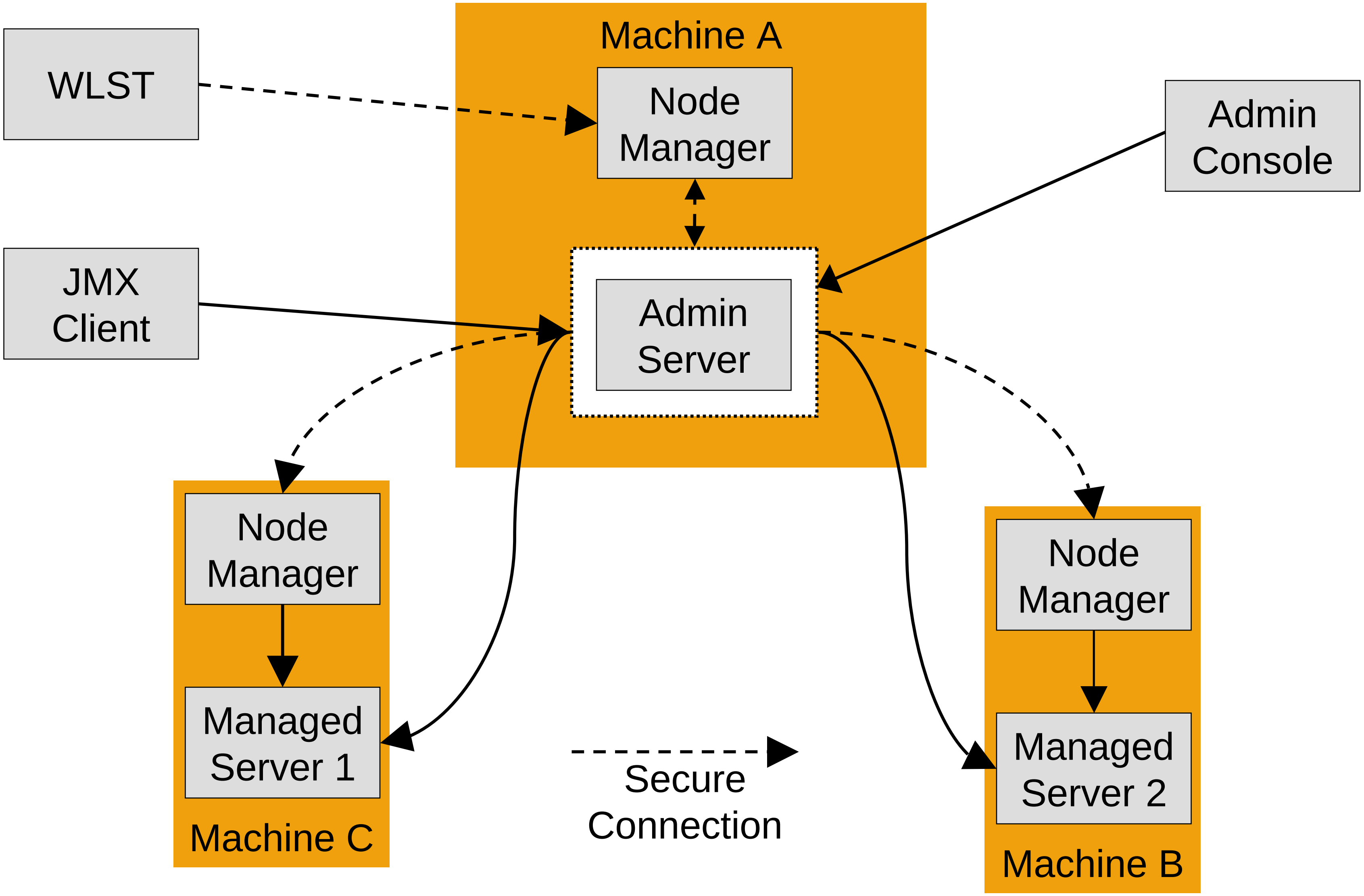
Node Manager

- Node Manager (NM):
  - Lets you start and kill managed servers remotely: one server, a domain, a cluster
  - Is available as either a Java-based or (for UNIX or Linux) a script-based process.
  - Monitors and acts on server health
  - Runs on the same computers as the managed servers
  - Can be run automatically in the background, as a Windows service or a Unix daemon

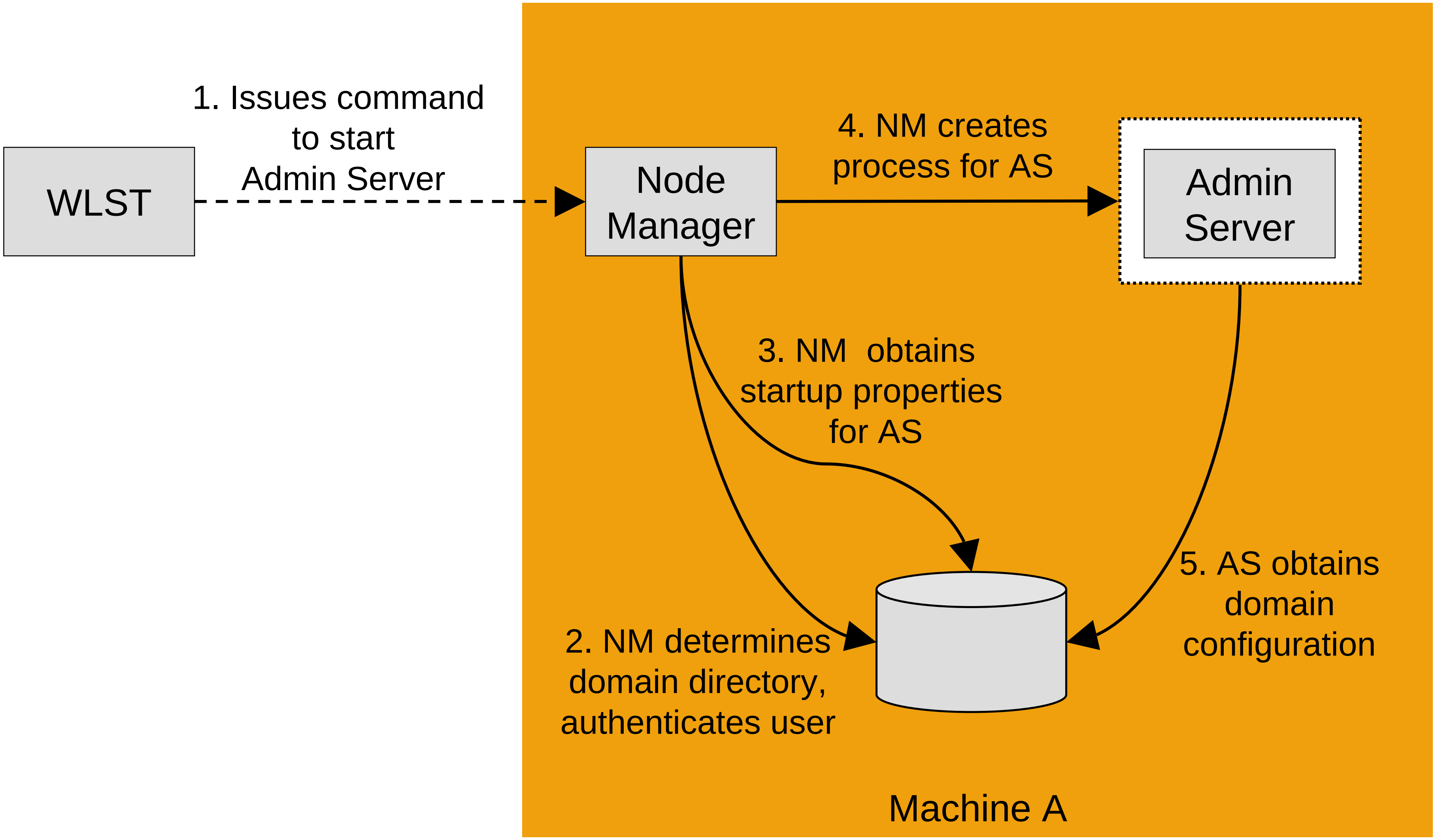
What Node Manager Can Do

- Node Manager can be used to:
  1. Start, Shut Down, and Restart an Administration Server.
  2. Start, Shut Down, Suspend, and Restart Managed Servers.
  3. Automatically Restart Administration and Managed Servers on failure.
  4. Monitor Servers and collects log data.

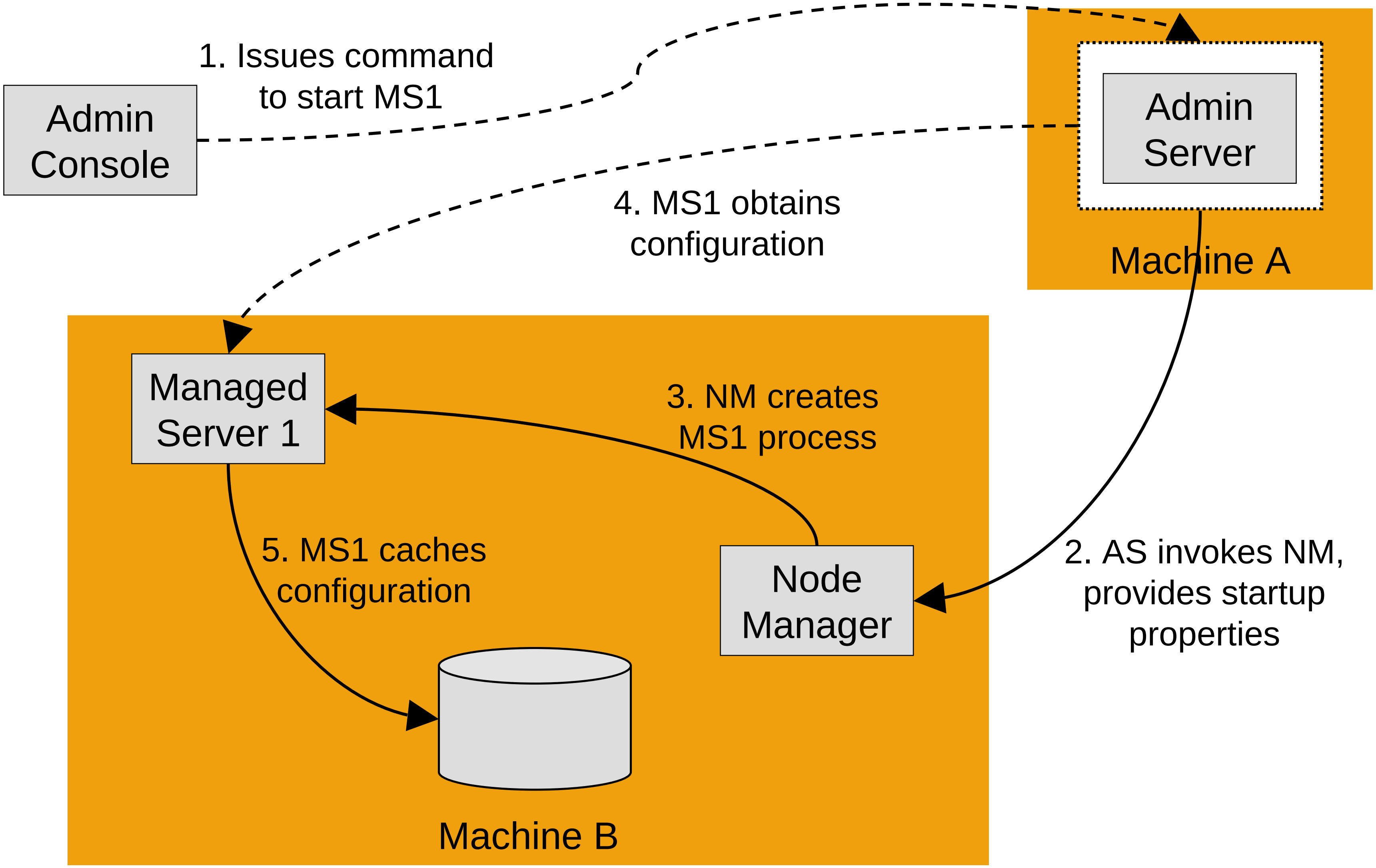
Node Manager Architecture



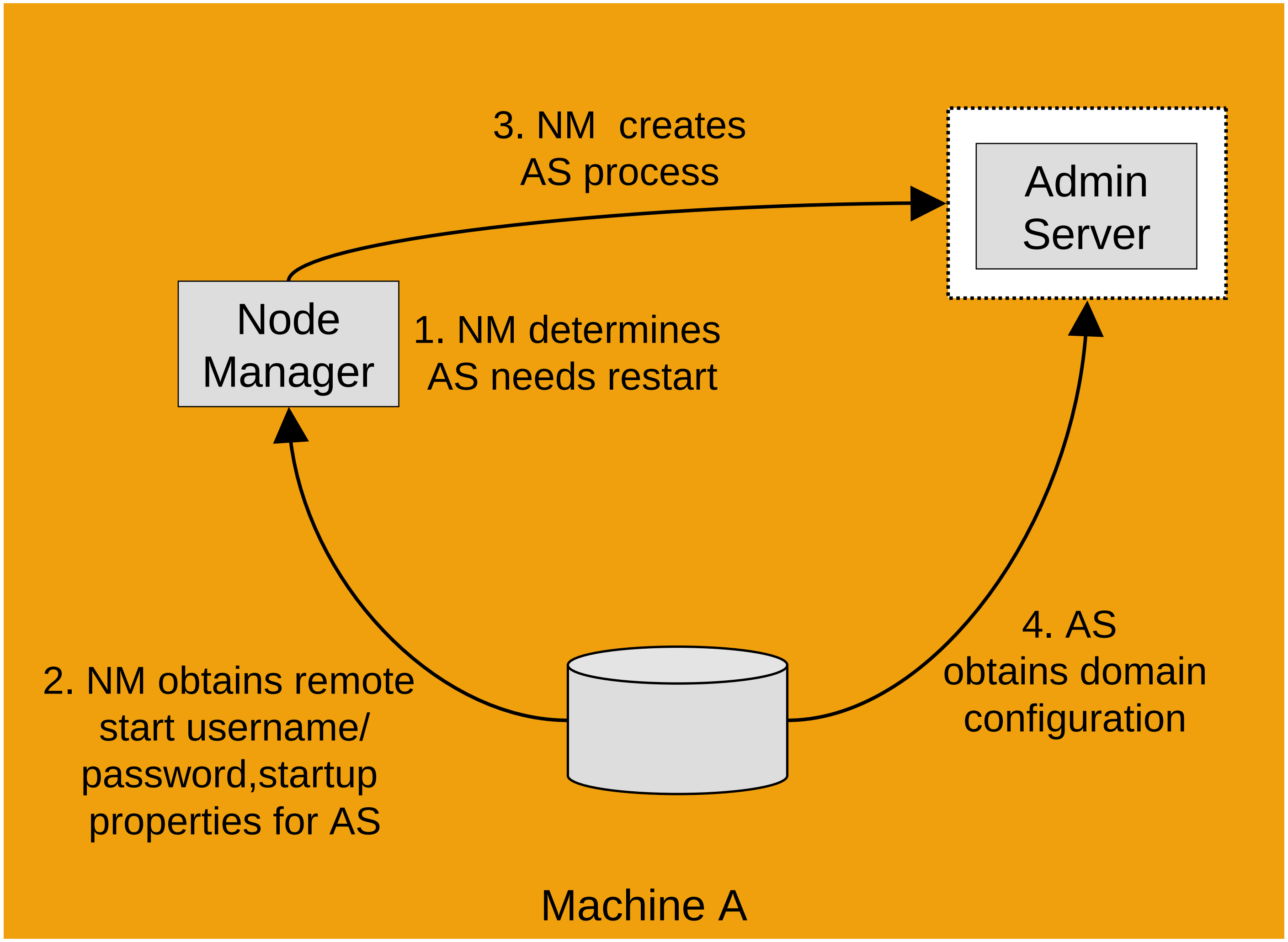
How Node Manager Starts an Administration Server


1. Issues command to start Admin Server
WLST
Node Manager
4. NM creates process for AS
Admin Server
3. NM obtains startup properties for AS
5. AS obtains domain configuration
2. NM determines domain directory, authenticates user
Machine A

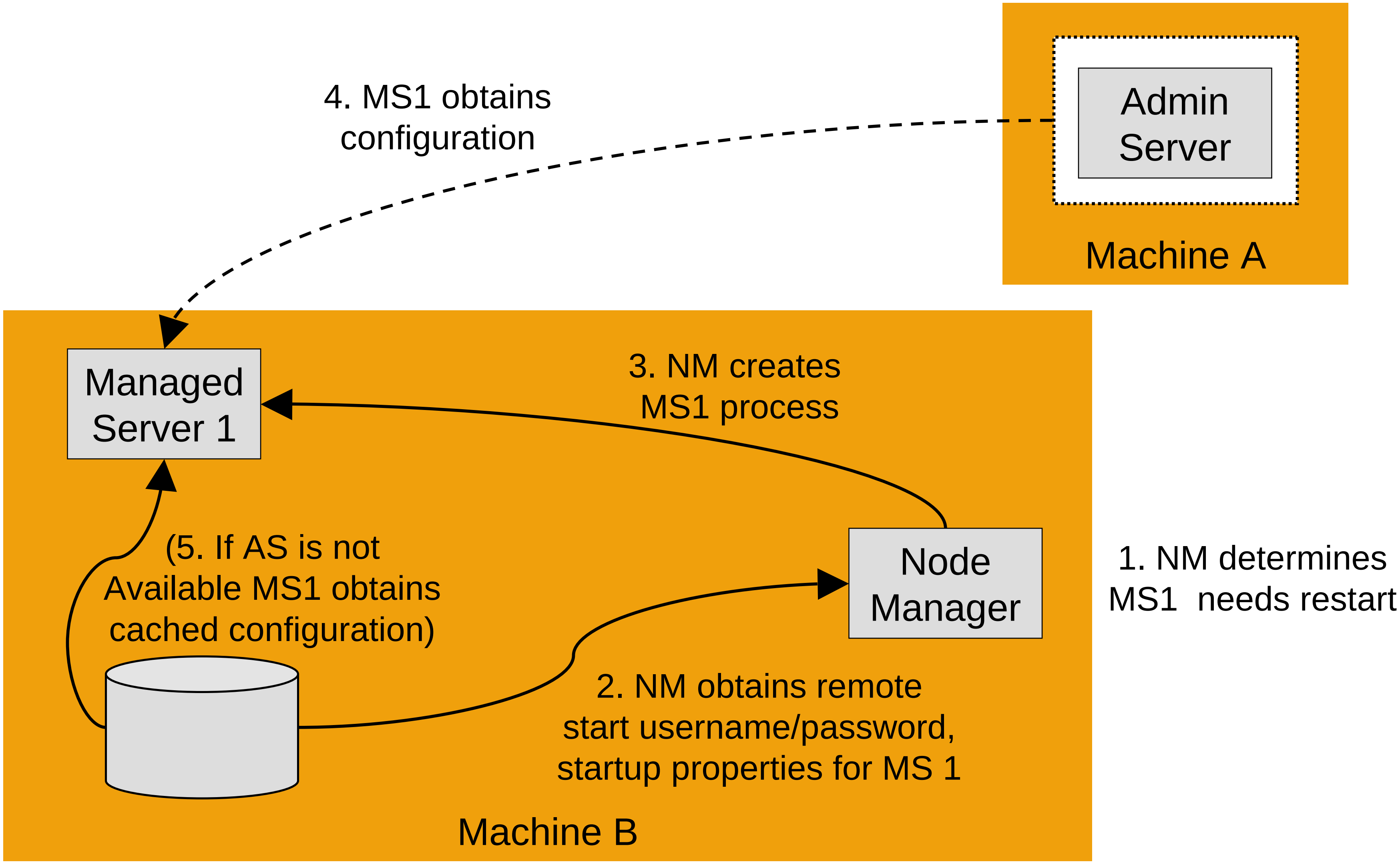
How Node Manager Starts a Managed Server


Admin Console
1. Issues command to start MS1
Admin Server
Machine A
4. MS1 obtains configuration
Managed Server 1
3. NM creates MS1 process
5. MS1 caches configuration
Node Manager
2. AS invokes NM, provides startup properties
Machine B

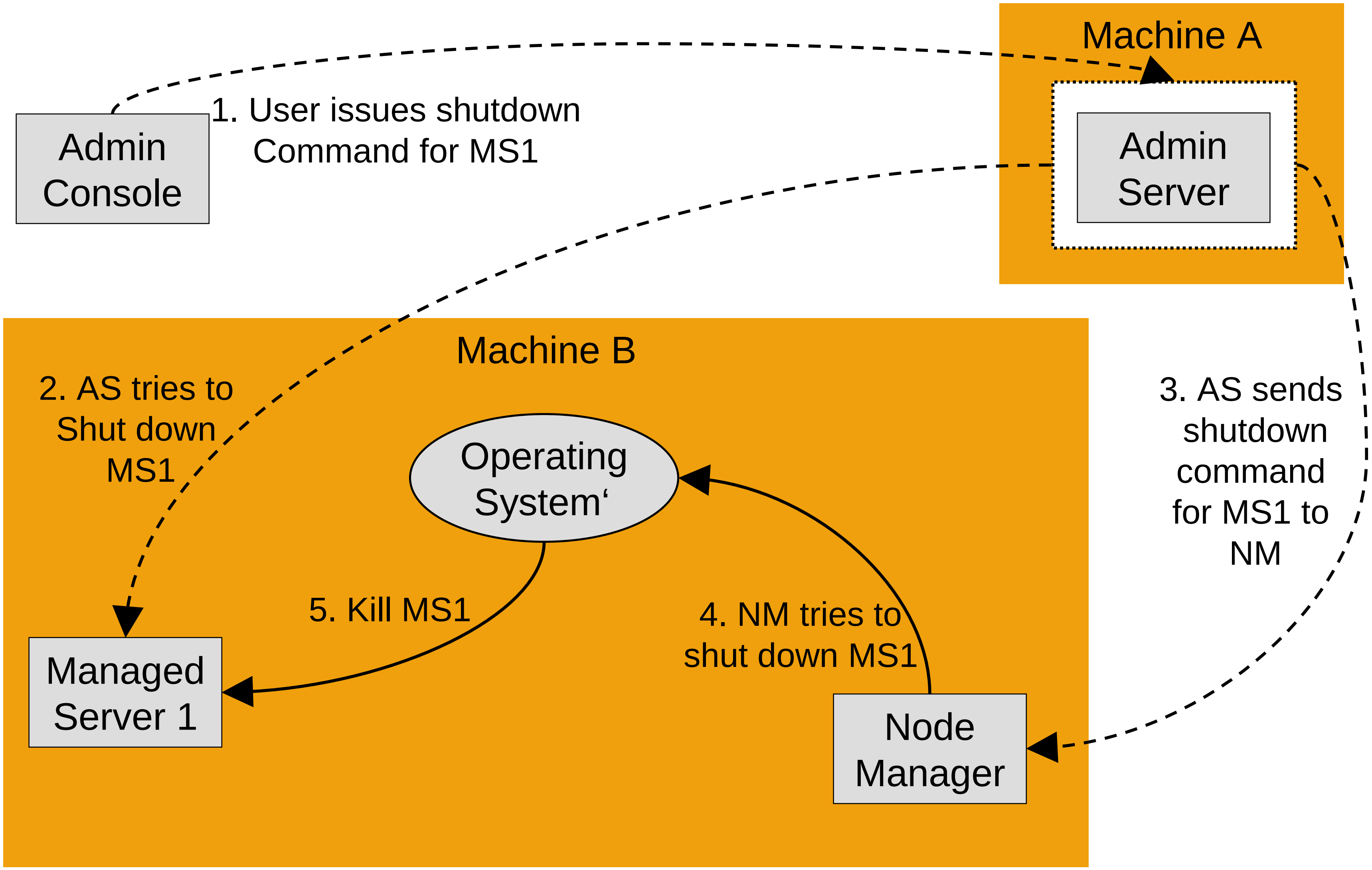
How Node Manager Restarts an Administration Server


3. NM creates AS process
Admin Server
Node Manager
1. NM determines AS needs restart
2. NM obtains remote start username/ password,startup properties for AS
4. AS obtains domain configuration
Machine A

How Node Manager Restarts a Managed Server


Admin Server
Machine A
4. MS1 obtains configuration
Managed Server 1
3. NM creates MS1 process
Node Manager
1. NM determines MS1 needs restart
(5. If AS is not Available MS1 obtains cached configuration)
2. NM obtains remote start username/password, startup properties for MS 1
Machine B

How Node Manager Shuts Down a Server Instance


Machine A
Admin Server
Admin Console
1. User issues shutdown Command for MS1
Machine B
2. AS tries to Shut down MS1
Operating System‘
3. AS sends shutdown command for MS1 to NM
5. Kill MS1
4. NM tries to shut down MS1
Managed Server 1
Node Manager

Versions of Node Manager

- There are two versions of Node Manager
  1. Java-Based Node Manager
  2. Script-Based Node Manager
- Java-based Node Manager runs within a Java Virtual Machine (JVM) process
- Script-based Node Manager (used only for UNIX and Linux systems)
  - Script-based does not have as much security, but provides the ability to remotely manage servers over a network using Secure Shell (SSH).

Node Manager Configuration

- NM must run on each computer that hosts WLS instances that you want to control with NM
- Configure each computer as a machine in WLS, and assign each server instance to be controlled by NM to the machine it runs on.
- NM should run as an operating system service, so that it automatically restarts upon system failure or reboot

Node Manager Default Behaviors

- After WebLogic Server installation, Node Manger is “ready-to-run” if the Node Manager and Administration Server are on the same machine.
- By default, the following behaviors are configured:
  - Administration console can use the Node Manager to start managed servers
  - Node Manager monitors the Managed Servers that it started
  - Automatic restart of Managed Servers is enabled

Configuring Java-Based Node Manager

- BEA recommends configuring NM to run as an operating system service
- Configuration tasks for Java-based Node Manager include:
  - Reconfiguring Startup Service for Windows Installation
  - Daemonizing Node Manager for UNIX systems
  - Configuring Java-based Node Manager Security
  - Reviewing `nodemanager.properties`
  - Configuring Node Manager on Multiple Machines

Reconfigure Startup Service for Windows Installation

1. Delete the service using `uninstallNodeMgrSvc.cmd`
2. Edit `installNodeMgrSvc.cmd` to specify NM’s Listen Address and Listen Port
3. Run `installNodeMgrSvc.cmd` to reinstall NM as a service, listening on the updated address and port

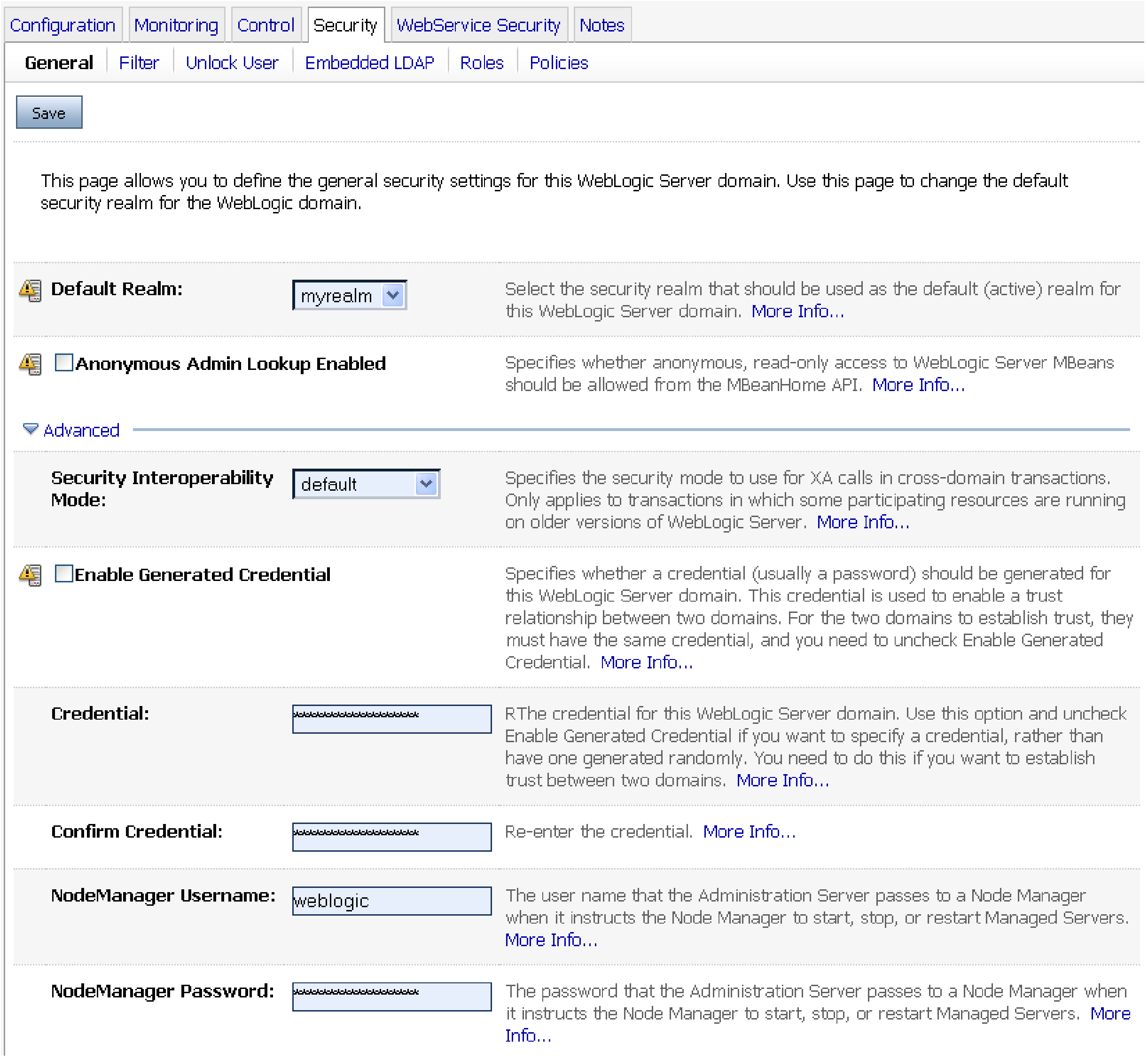
Daemonizing NM for UNIX Systems

1. Remove NM daemon process setup from WLS installation
2. Reinstall NM daemon
3. Configure NM:
   - Set `WL_HOME`
   - Set `NODEMGR_HOME`
   - Add JDK and WL directories to system path
   - Add JDK and WL jars to classpath
   - Set `LD_LIBRARY_PATH`
   - Set `JAVA_VM`
   - Set `NODEMGR_HOST`
   - Set `NODEMGR_PORT`
   - Set `PROD_NAME=`BEA WebLogic Platform 9.1

Configuring Java-Based Node Manager Security

- NM Security relies on a one-way SSL connection between client and server
- WLST uses the `nmConnect` command to establish a connection to the Java Node Manager.
- The `nmConnect` command requires a username and password, which is verified against the `nm_password.properties` file.

Administration Console NM Security



This page allows you to define the general security settings for this WebLogic Server domain. Use this page to change the default security realm for the WebLogic domain.

Default Realm:	myrealm	Select the security realm that should be used as the default (active) realm for this WebLogic Server domain. More Info...
□ **Anonymous Admin Lookup Enabled**		Specifies whether anonymous, read-only access to WebLogic Server MBeans should be allowed from the MBeanHome API. More Info...
Security Interoperability Mode:	default	Specifies the security mode to use for XA calls in cross-domain transactions. Only applies to transactions in which some participating resources are running on older versions of WebLogic Server. More Info...
□ **Enable Generated Credential**		Specifies whether a credential (usually a password) should be generated for this WebLogic Server domain. This credential is used to enable a trust relationship between two domains. For the two domains to establish trust, they must have the same credential, and you need to uncheck Enable Generated Credential. More Info...
Credential:	******************	RThe credential for this WebLogic Server domain. Use this option and uncheck Enable Generated Credential if you want to specify a credential, rather than have one generated randomly. You need to do this if you want to establish trust between two domains. More Info...
Confirm Credential:	******************	Re-enter the credential. More Info...
NodeManager Username:	weblogic	The user name that the Administration Server passes to a Node Manager when it instructs the Node Manager to start, stop, or restart Managed Servers. More Info...
NodeManager Password:	******************	The password that the Administration Server passes to a Node Manager when it instructs the Node Manager to start, stop, or restart Managed Servers. More Info...

Remote Server Start Security for Java-Based Node Manager

- Credentials for Managed Servers and Administration Servers are handled differently
  - Managed Servers – When you invoke NM to start a Managed Server it gets its remote username and password from the Administration Server
  - Administration Servers – When you invoke NM to start an Administration Server, the remote start username come from either the command-line or the `boot.properties` file

Reviewing `nodemanager.properties`

- Properties for a Java-based Node Manager process can be specified either at the command line or in the `nodemanager.properties` file.
- Values supplied on the command line take precedence over those in the `nodemanager.properties` file.
- To configure the Node Manager to use a start script, in the nodemanager.properties file:
  1. set the StartScriptEnabled property to true.
  2. Set the StartScriptName property to the name of your script

Configuring Node Manager on Multiple Machines

- Node Manager has to be installed and configured on each machine on which there is a Managed Server
- This can be done with the WLST `nmEnroll` command to copy all required domain and configuration information from one machine to another.

Configurating Script-Based Node Manager

- The SSH Node Manager is a shell script, wlscontrol.sh, located in NM_HOME/.
- An executable SSH client must reside on each machine where Node Manager or Node Manager client runs.
  - An SSH client is typically a standard part of a Unix or Linux installation
- Configuration tasks for Script-based Node Manager include:
  - Using SSL With Script-based Node Manager
  - Creating a Node Manager User
  - Configuring Script-based Node Manager Security

Using SSL With Script-based NM

- Script-based Node Manager communicates with Administration Servers and Managed Servers using one-way SSL.
- The default WLS installation includes demonstration Identity and Trust keystores that allow SSL to be used out of the box.
- To configure SSL for the production environment, identity and trust must be obtained for the Node Manager, the Administration Server and all Managed Servers.

Creating a Node Manager User

- Before running Node Manager, a dedicated UNIX user account – for performing Node Manager functions – should be created.
- This user should be added to all machines that will host the SSH Node Manager and to all machines that will host a Node Manager client, including the Administration Server.

Configuring Script-Based Node Manager Security

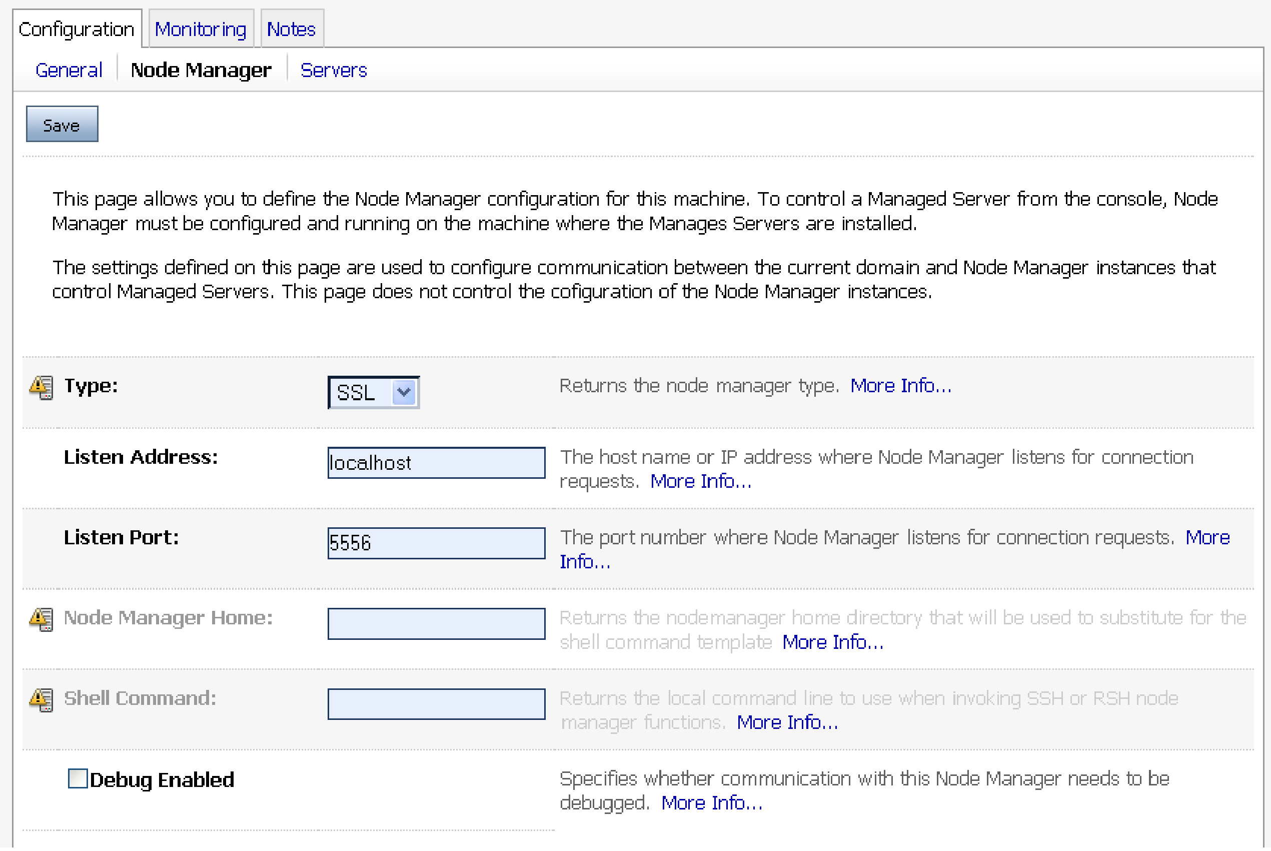
- The Node Manager SSH shell script relies on SSH user-based security to provide a secure trust relationship between users on different machines.
- Authentication is not required.
- You create a UNIX user account – typically one per domain – for running Node Manager commands and scripts.
- A user logged in as this user can issue Node Manager commands without providing a username and password.

Additional Configuration Information

- Other Node Manager configuration tasks include:
  - Configuring a Machine to User Node Manager
  - Configuring `nodemanager.domains` file
  - Configuring Remote Startup Arguments
  - Ensuring Administration Server Address is Defined
  - Setting Node Manager Environment Variables

Configuring a Machine to User Node Manager

- A WLS Machine resource maps a machine with the server instances it hosts.



This page allows you to define the Node Manager configuration for this machine. To control a Managed Server from the console, Node Manager must be configured and running on the machine where the Manages Servers are installed.

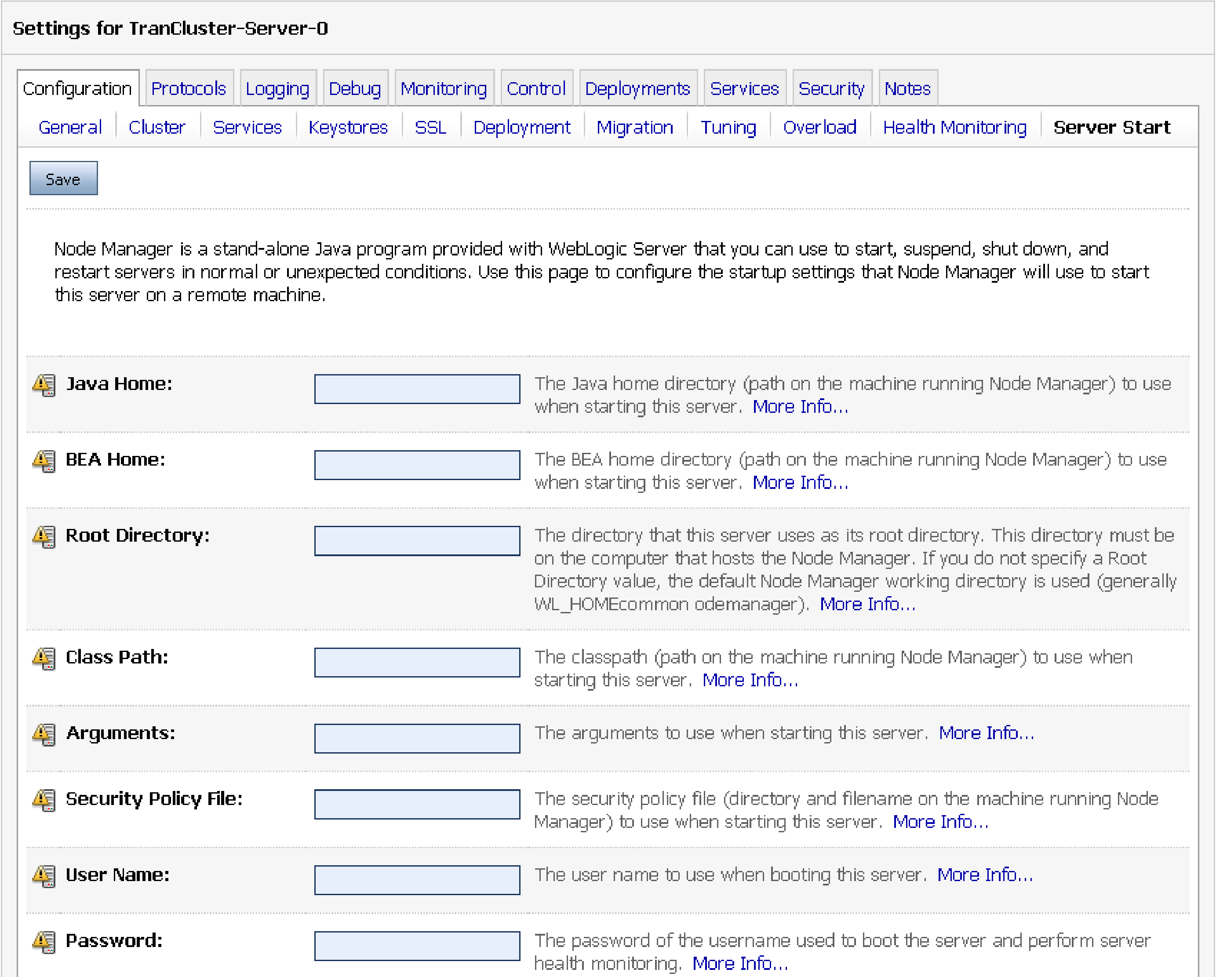
The settings defined on this page are used to configure communication between the current domain and Node Manager instances that control Managed Servers. This page does not control the cofiguration of the Node Manager instances.

Type:	SSL	Returns the node manager type. More Info...
Listen Address:	localhost	The host name or IP address where Node Manager listens for connection requests. More Info...
Listen Port:	5556	The port number where Node Manager listens for connection requests. More Info...
Node Manager Home:		Returns the nodemanager home directory that will be used to substitute for the shell command template More Info...
Shell Command:		Returns the local command line to use when invoking SSH or RSH node manager functions. More Info...
□ **Debug Enabled**		Specifies whether communication with this Node Manager needs to be debugged. More Info...

Configuring nodemanager.domains File

- The `nodemanager.domains` file specifies the domains that a Node Manager instance controls.
- When a user issues a command for a domain, NM looks up the domain directory from this file.
- `nodemanager.domains` provides additional security by restricting Node Manager client access to the domains listed in this file.

Configuring Remote Startup Arguments



Settings for TranCluster-Server-0

Node Manager is a stand-alone Java program provided with WebLogic Server that you can use to start, suspend, shut down, and restart servers in normal or unexpected conditions. Use this page to configure the startup settings that Node Manager will use to start this server on a remote machine.

Java Home: The Java home directory (path on the machine running Node Manager) to use when starting this server. More Info...

BEA Home: The BEA home directory (path on the machine running Node Manager) to use when starting this server. More Info...

Root Directory: The directory that this server uses as its root directory. This directory must be on the computer that hosts the Node Manager. If you do not specify a Root Directory value, the default Node Manager working directory is used (generally WL_HOMEcommon odemanager). More Info...

Class Path: The classpath (path on the machine running Node Manager) to use when starting this server. More Info...

Arguments: The arguments to use when starting this server. More Info...

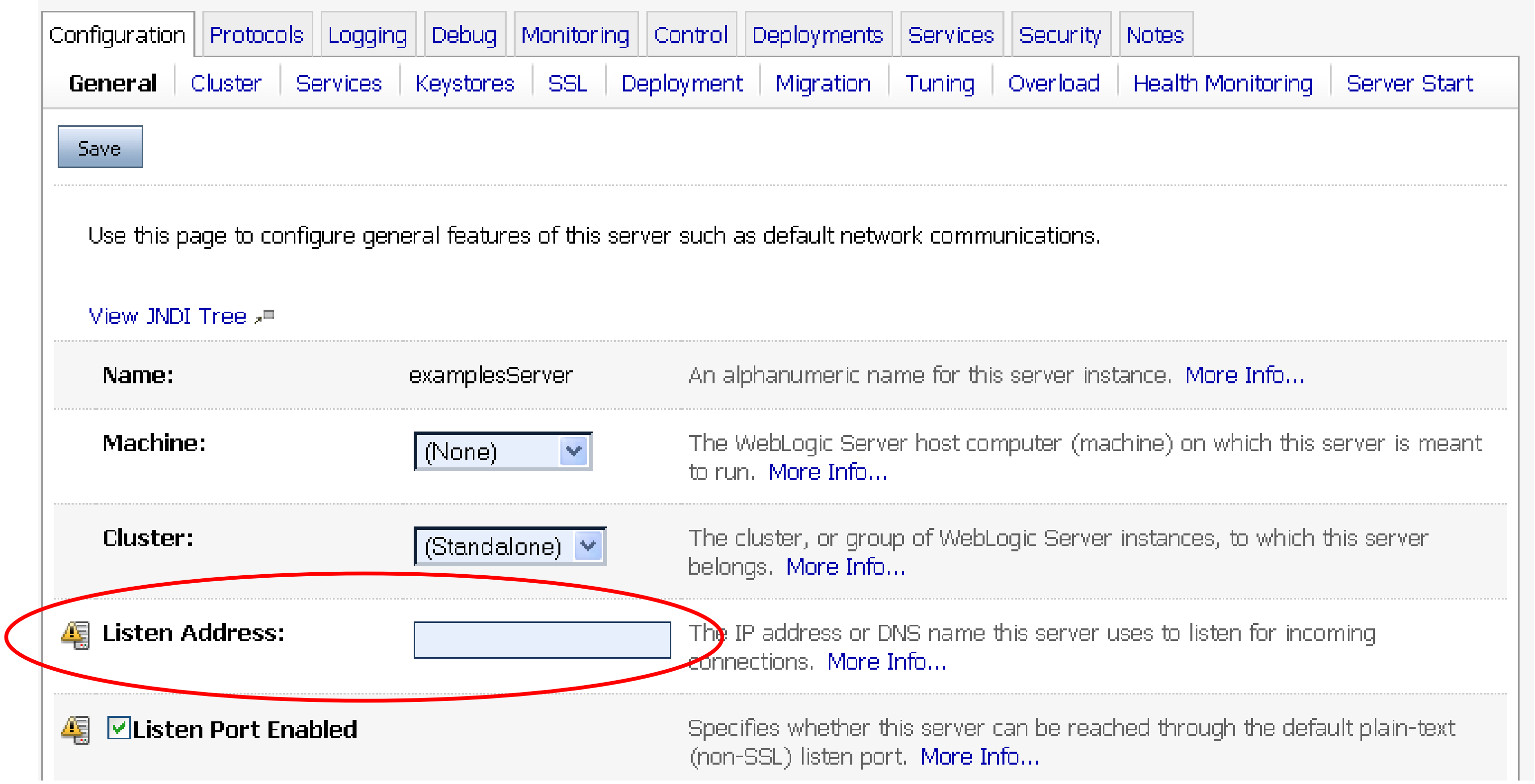
Security Policy File: The security policy file (directory and filename on the machine running Node Manager) to use when starting this server. More Info...

User Name: The user name to use when booting this server. More Info...

Password: The password of the username used to boot the server and perform server health monitoring. More Info...

Ensuring Administration Server Address is Defined

- A Listen Address must be defined for each Administration Server that will connect to the Node Manager process.



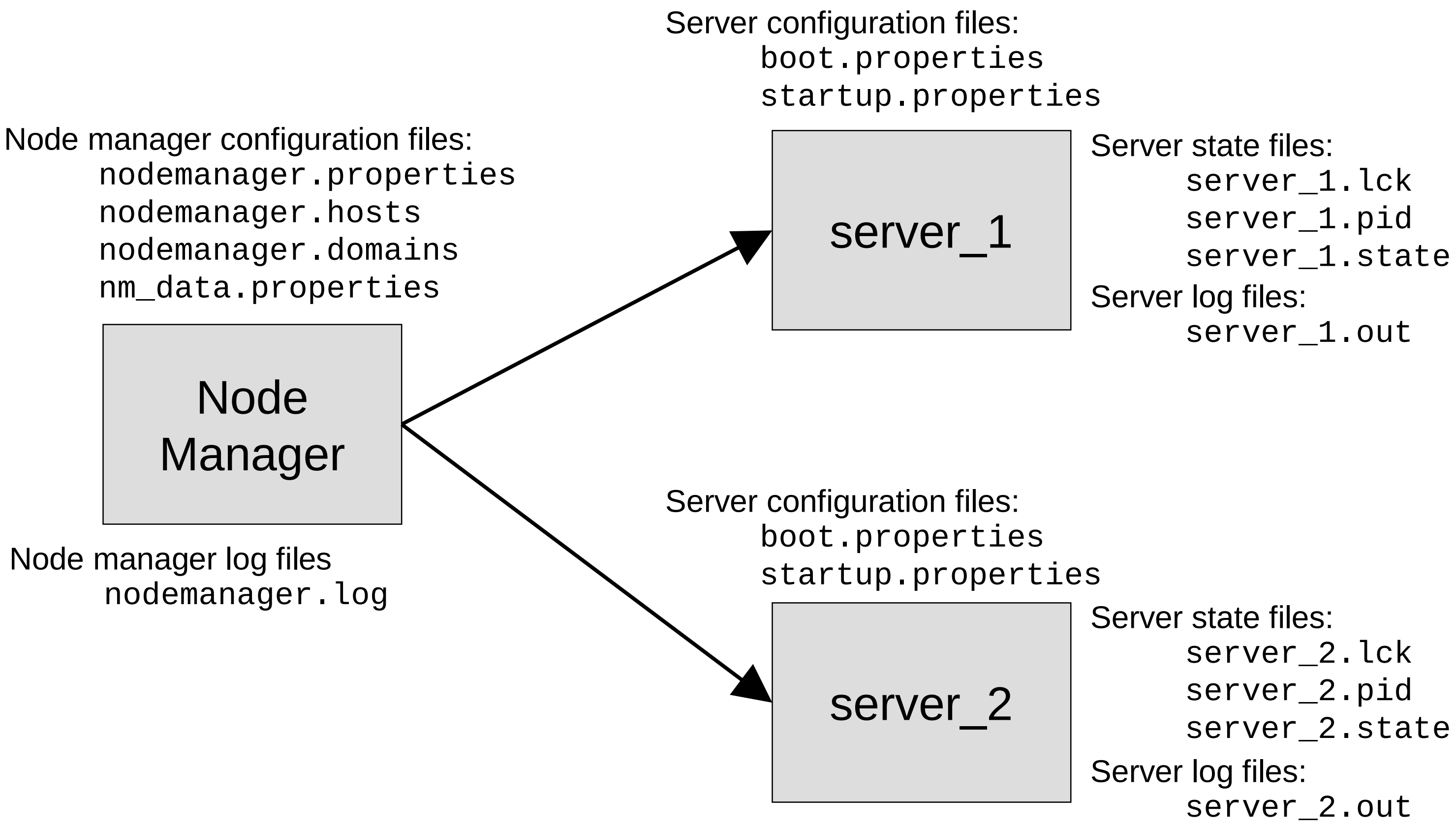
Use this page to configure general features of this server such as default network communications.

Name:	examplesServer	An alphanumeric name for this server instance. More Info...
Machine:	(None)	The WebLogic Server host computer (machine) on which this server is meant to run. More Info...
Cluster:	(Standalone)	The cluster, or group of WebLogic Server instances, to which this server belongs. More Info...
Listen Address:		The IP address or DNS name this server uses to listen for incoming connections. More Info...
☑ Listen Port Enabled		Specifies whether this server can be reached through the default plain-text (non-SSL) listen port. More Info...

Setting Node Manager Environment Variables

Environment Variable	Description
JAVA_HOME	Root directory of JDK that you are using for Node Manager. For example: set JAVA_HOME=c:\bea\jdk1.5.0_04 Node Manager has the same JDK version requirements as WebLogic Server.
WL_HOME	WebLogic Server installation directory. For example: set WL_HOME=c:\bea\weblogic91
PATH	Must include the WebLogic Server bin directory and path to your Java executable. For example: set PATH=%WL_HOME%\server\bin;%JAVA_HOME%\bin;%PATH%
LD_LIBRARY_PATH (UNIX only)	For HP UX and Solaris systems, you must include the path to the native Node Manager libraries. Solaris example: LD_LIBRARY_PATH:$WL_HOME/server/lib/solaris:$WL_HOME/server/lib/solaris/oci816_8 HP UX example: SHLIB_PATH=$SHLIB_PATH:$WL_HOME/server/lib/hpux11:$WL_HOME/server/lib/hpux11/oci816_8
CLASSPATH	You can set the Node Manager CLASSPATH either as an option on the java command line used to start Node Manager, or as an environment variable. Windows example: set CLASSPATH=.;%WL_HOME%\server\lib\weblogic_sp.jar;%WL_HOME%\server\lib\weblogic.jar

Node Manager Configuration and Log Files



Node manager configuration files:

```
nodemanager.properties
nodemanager.hosts
nodemanager.domains
nm_data.properties
```

Node Manager

Node manager log files

```
nodemanager.log
```

Server configuration files:

```
boot.properties
startup.properties
```

server_1

Server state files:

```
server_1.lck
server_1.pid
server_1.state
```

Server log files:

```
server_1.out
```

Server configuration files:

```
boot.properties
startup.properties
```

server_2

Server state files:

```
server_2.lck
server_2.pid
server_2.state
```

Server log files:

```
server_2.out
```

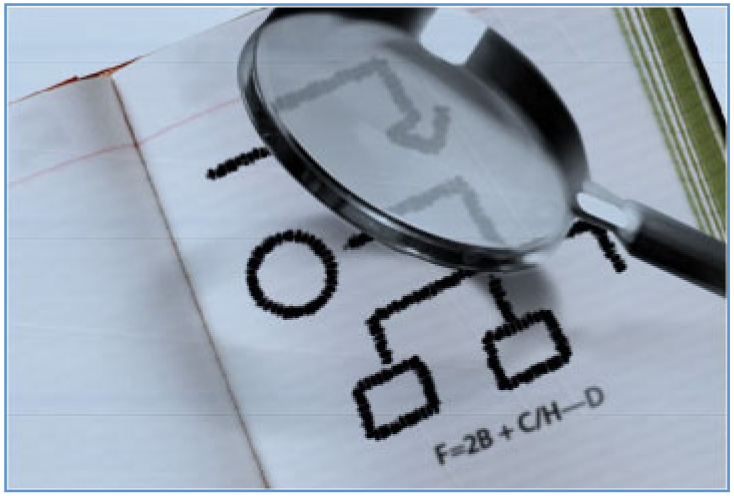
Node Manager Configuration and Log Files

- Node Manager config files include:
  - `nodemanager.properties`
  - `nodemanager.hosts`
  - `nodemanager.domains`
  - `nm_data.properties`
  - `nm_password.properties`
  - `boot.properties`
  - `startup.properties`
  - `server_name.lck`
  - `server_name.pid`
  - `server_name.state`
- Node Manager log files include:
  - `nodemanager.log`
  - `server_name.out`

Section Review

In this section we discussed:

- ✓ How to create a machine definition
- ✓ Targeting servers to a machine
- ✓ The benefits of Node Manager
- ✓ The five steps to setting it up
- ✓ Console operations made available by Node Manager



Exercise

Configuring Servers and Machines

- In this lab you are going to create and configure two machines.
- Ask the instructor for any clarification.
- The instructor will determine the stop time.

Lab Exercise

Exercise

Starting Servers Using Node Manager

- In this lab you will use Node Manager to control managed servers.
- Ask the instructor for any clarification.
- The instructor will determine the stop time.

Lab Exercise

Road Map

1. Remote Administration
2. **Logs and Monitoring**
   - Using Log Files
   - Monitoring Servers
3. SNMP Concepts
4. WLS SNMP Agent
5. WLS SNMP Management Tools
6. Network Channels

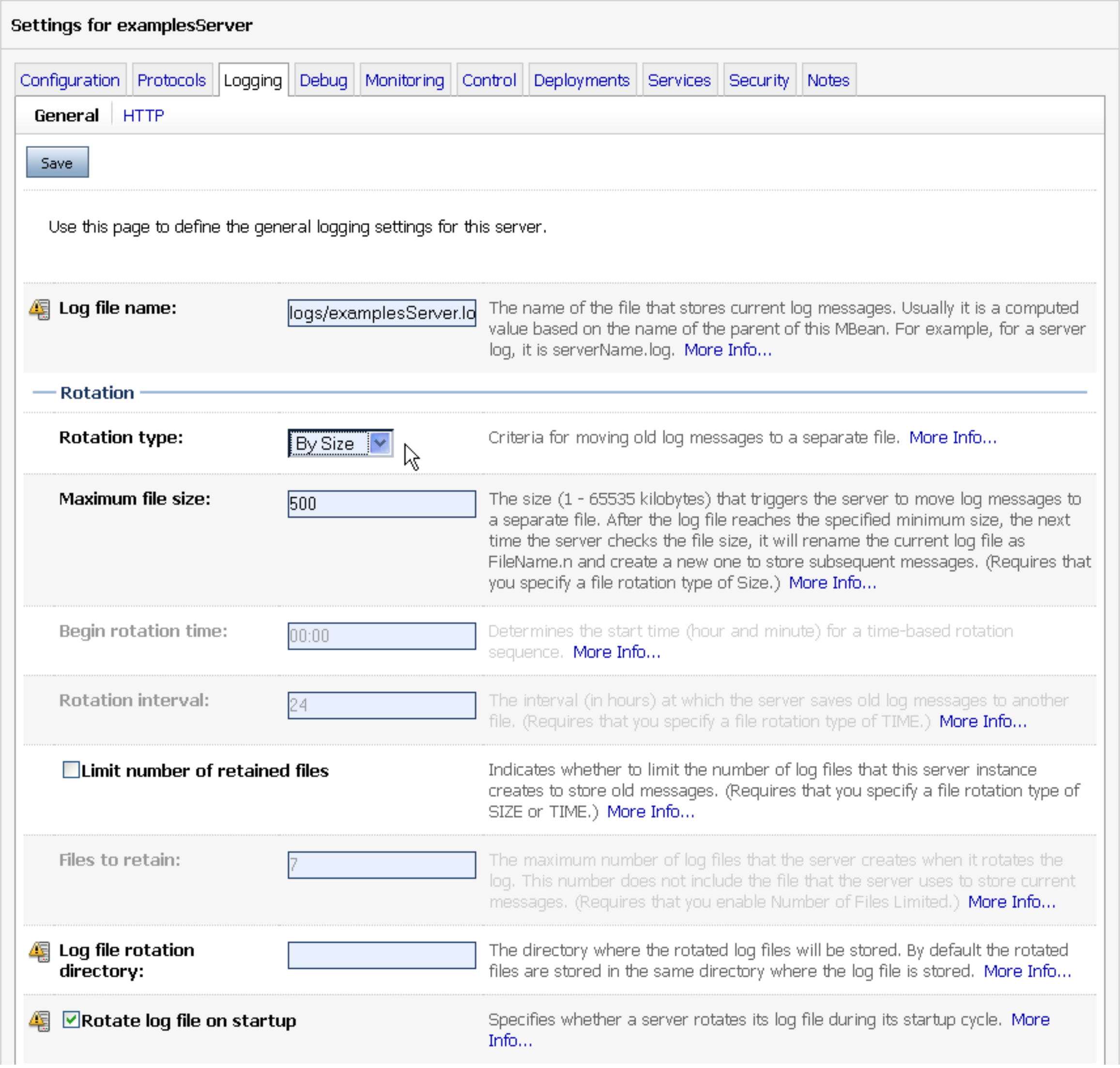
Using Logs

- Logs can aid in the discovery of:
  - frequently accessed resources
  - activity by day and time interval
  - amount of data sent and received
  - IP addresses of users accessing the site
  - number of actual “hits”
  - problems servicing requests
  - performance statistics

Main Server Logs

- A Server log:
  - logs all server activity
  - is stored in `serverName\logs\<serverName>.log` by default
- A Domain log:
  - logs all domain activity
  - is stored in `<AdminServer>\logs\<domainName>.log` by default
- These logs are independently configured.

Configuring Server Logging



Settings for examplesServer

Use this page to define the general logging settings for this server.

Log file name: logs/examplesServer.lo — The name of the file that stores current log messages. Usually it is a computed value based on the name of the parent of this MBean. For example, for a server log, it is serverName.log. More Info...

Rotation

Rotation type: By Size — Criteria for moving old log messages to a separate file. More Info...

Maximum file size: 500 — The size (1 - 65535 kilobytes) that triggers the server to move log messages to a separate file. After the log file reaches the specified minimum size, the next time the server checks the file size, it will rename the current log file as FileName.n and create a new one to store subsequent messages. (Requires that you specify a file rotation type of Size.) More Info...

Begin rotation time: 00:00 — Determines the start time (hour and minute) for a time-based rotation sequence. More Info...

Rotation interval: 24 — The interval (in hours) at which the server saves old log messages to another file. (Requires that you specify a file rotation type of TIME.) More Info...

□ **Limit number of retained files** — Indicates whether to limit the number of log files that this server instance creates to store old messages. (Requires that you specify a file rotation type of SIZE or TIME.) More Info...

Files to retain: 7 — The maximum number of log files that the server creates when it rotates the log. This number does not include the file that the server uses to store current messages. (Requires that you enable Number of Files Limited.) More Info...

Log file rotation directory: — The directory where the rotated log files will be stored. By default the rotated files are stored in the same directory where the log file is stored. More Info...

☑ **Rotate log file on startup** — Specifies whether a server rotates its log file during its startup cycle. More Info...

Messages Forwarded to Domain Log

Severity	Forwarded to Domain Log by Default	Meaning
Informational	No	Used for reporting normal operations.
Notice	Yes	An informational message with a higher level of importance
Warning	Yes	A suspicious operation or configuration has occurred but it may not have an impact on normal operation.
Error	Yes	A user error has occurred. The system or application is able to handle the error with no interruption, and limited degradation, of service.
Critical	Yes	A system or service error has occurred. The system is able to recover but there might be a momentary loss, or permanent degradation, of service.
Alert	Yes	A particular service is in an unusable state while other parts of the system continue to function. Automatic recovery is not possible; the immediate attention of the administrator is needed to resolve the problem.
Emergency	Yes	The server is in an unusable state. This severity indicates a severe system failure or panic.

Message Attributes

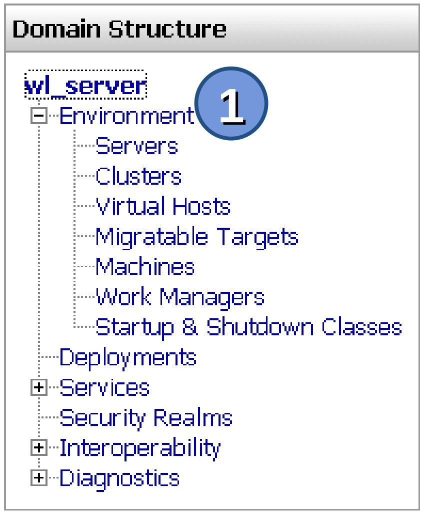
```
####<Jun 2, 2000 10:23:02 AM PDT> <Info> <SSL> <bigbox> <myServer>
<SSLListenThread> <harry> <> <004500> <Using exportable strength SSL>
```

Attribute	Description
Timestamp	The time and date when the message originated, in a format that is specific to the locale.
Severity	Indicates the degree of impact or seriousness of the event reported by the message.
Subsystem	This attribute denotes the particular subsystem of WebLogic Server that was the source of the message. For example, EJB, RMI, JMS.
Server Name Machine Name Thread ID Transaction ID	These four attributes identify the origins of the message. Transaction ID is present only for messages logged within the context of a transaction. Note: Server Name and Thread ID are not present in log messages generated by a Java client and logged to a client log.
User ID	The user from the security context when the message was generated.
Message ID	A unique six-digit identifier. Message IDs through 499999 are reserved for WebLogic Server system messages.
Message Text	For WebLogic Server messages, this contains the Short Description as defined in the system message catalog. For other messages, this is text defined by the developer of the program.

Log Filters

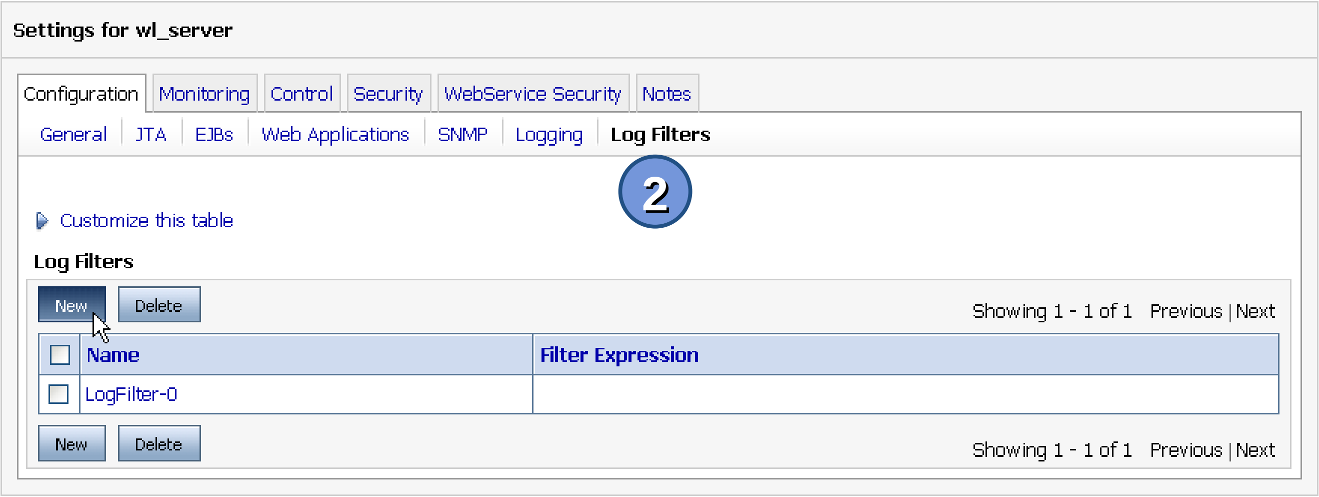
- Log filters provide control over the log messages that get published.
  - You can filter out messages of a certain severity level, from a particular subsystem, or according to specified criteria
- You can create separate filters for the messages that each server instance writes to:
  - its server log file
  - standard out
  - memory buffer
  - domain-wide log

Creating Log Filters



Domain Structure

- wl_server
  - ⊟ Environment
    - Servers
    - Clusters
    - Virtual Hosts
    - Migratable Targets
    - Machines
    - Work Managers
    - Startup & Shutdown Classes
  - Deployments
  - ⊞ Services
  - Security Realms
  - ⊞ Interoperability
  - ⊞ Diagnostics



Settings for wl_server

Configuration | Monitoring | Control | Security | WebService Security | Notes

General | JTA | EJBs | Web Applications | SNMP | Logging | **Log Filters**

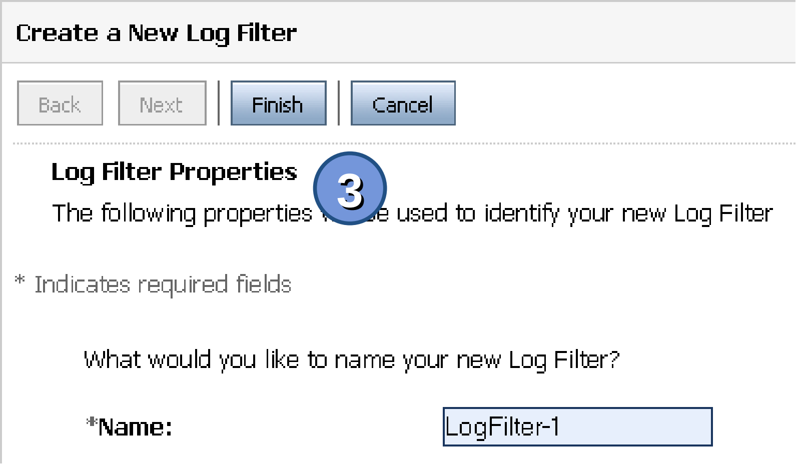
▷ Customize this table

Log Filters

New | Delete — Showing 1 - 1 of 1 Previous | Next

□	Name	Filter Expression
□	LogFilter-0	

New | Delete — Showing 1 - 1 of 1 Previous | Next



Create a New Log Filter

Back | Next | Finish | Cancel

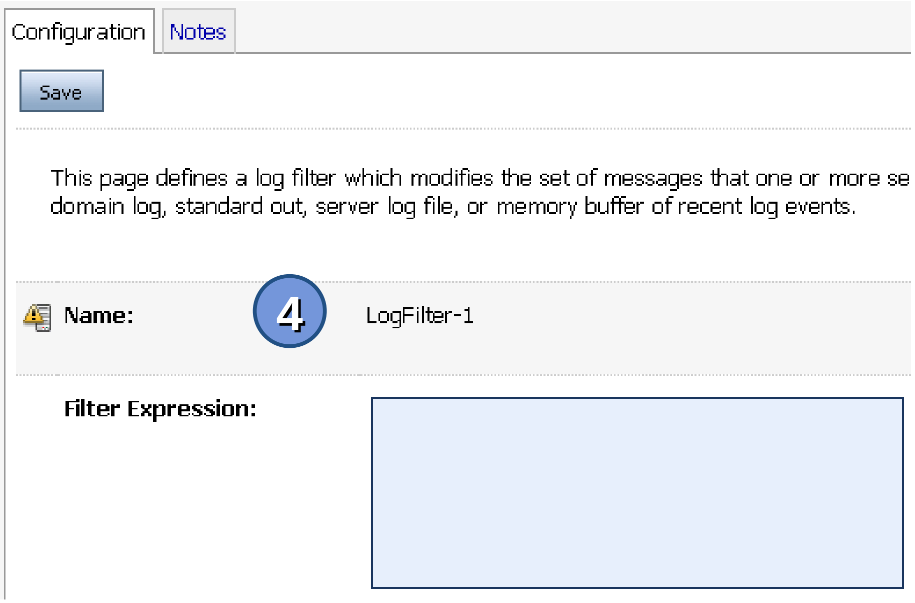
Log Filter Properties

The following properties will be used to identify your new Log Filter

* Indicates required fields

What would you like to name your new Log Filter?

***Name:** LogFilter-1



Configuration | Notes

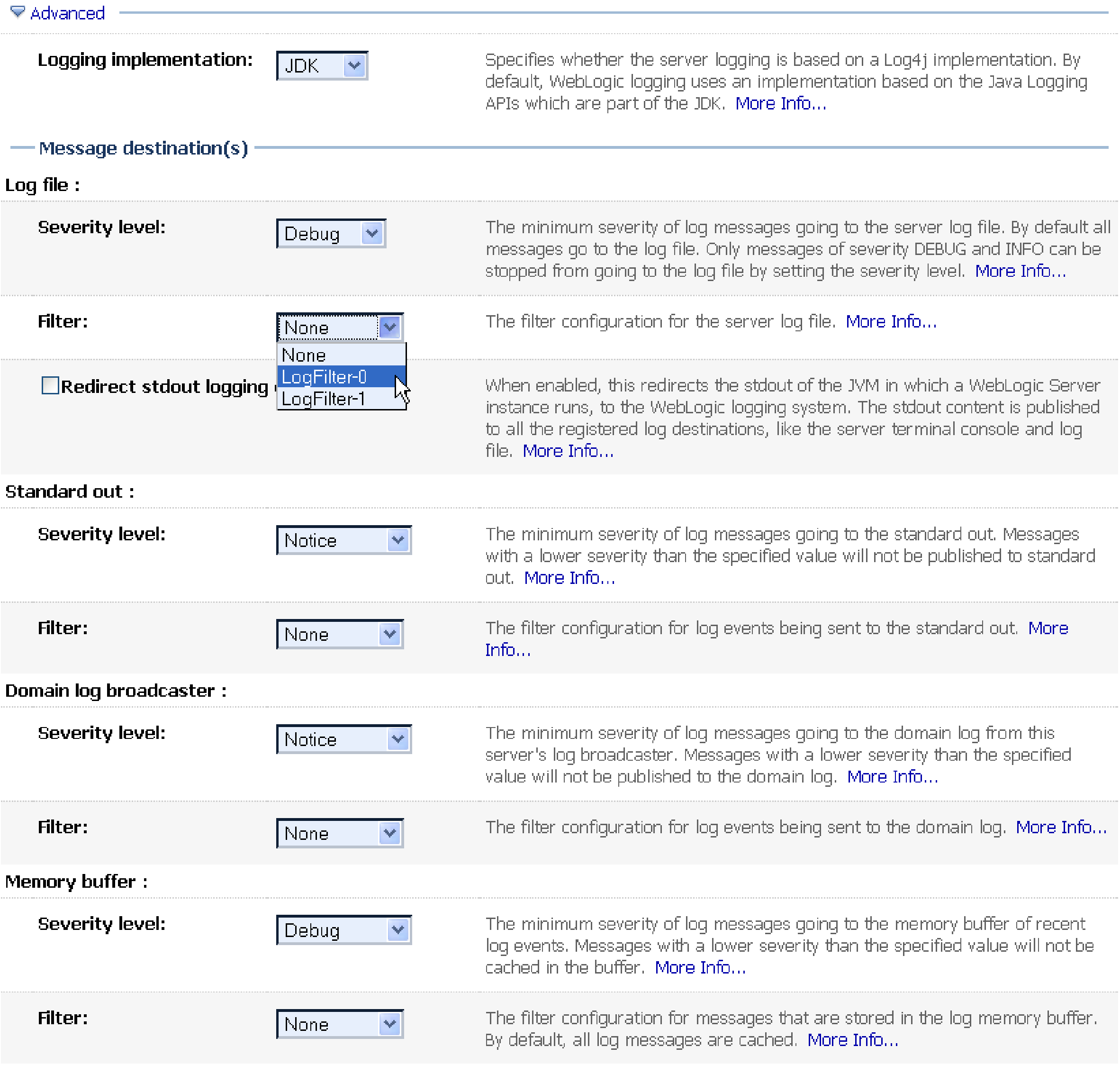
Save

This page defines a log filter which modifies the set of messages that one or more se domain log, standard out, server log file, or memory buffer of recent log events.

Name: LogFilter-1

Filter Expression:

Assigning a Log Filter



▼ Advanced

Logging implementation: JDK — Specifies whether the server logging is based on a Log4j implementation. By default, WebLogic logging uses an implementation based on the Java Logging APIs which are part of the JDK. More Info...

Message destination(s)

Log file :

Severity level: Debug — The minimum severity of log messages going to the server log file. By default all messages go to the log file. Only messages of severity DEBUG and INFO can be stopped from going to the log file by setting the severity level. More Info...

Filter: None — The filter configuration for the server log file. More Info...

☐ **Redirect stdout logging** — When enabled, this redirects the stdout of the JVM in which a WebLogic Server instance runs, to the WebLogic logging system. The stdout content is published to all the registered log destinations, like the server terminal console and log file. More Info...

Standard out :

Severity level: Notice — The minimum severity of log messages going to the standard out. Messages with a lower severity than the specified value will not be published to standard out. More Info...

Filter: None — The filter configuration for log events being sent to the standard out. More Info...

Domain log broadcaster :

Severity level: Notice — The minimum severity of log messages going to the domain log from this server's log broadcaster. Messages with a lower severity than the specified value will not be published to the domain log. More Info...

Filter: None — The filter configuration for log events being sent to the domain log. More Info...

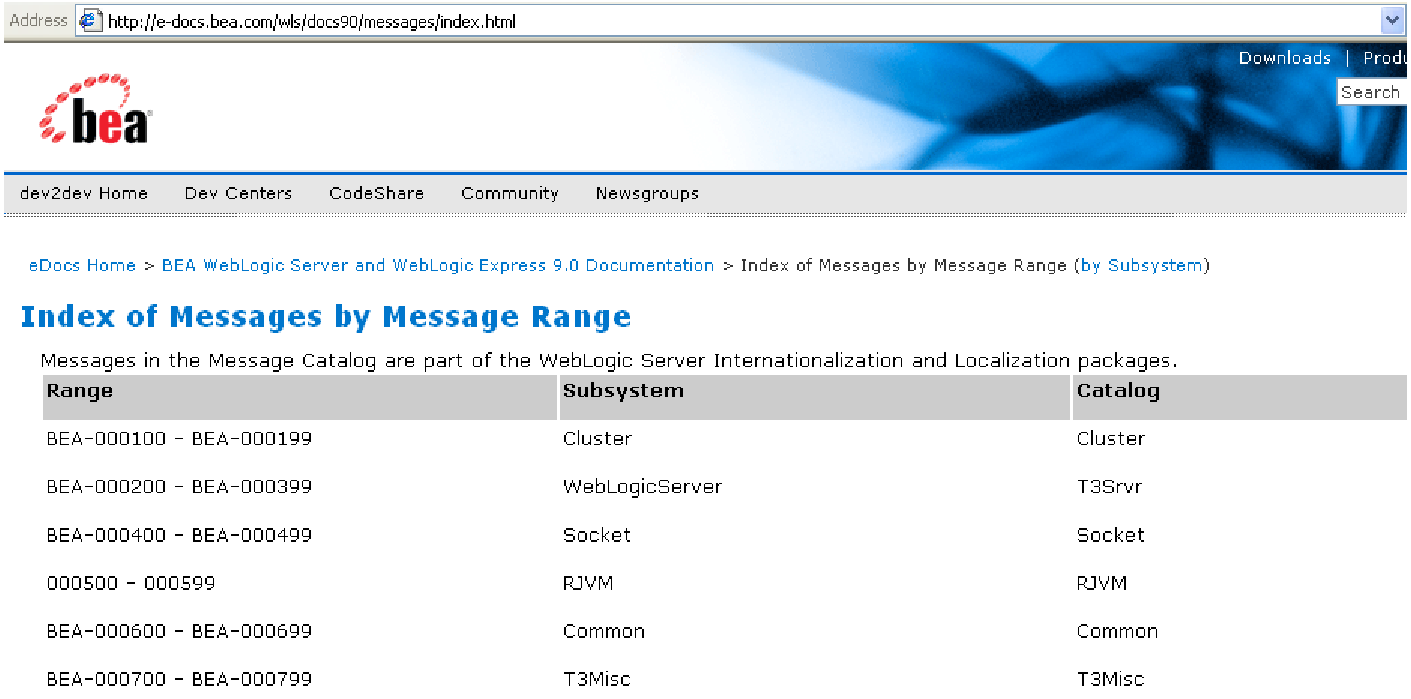
Memory buffer :

Severity level: Debug — The minimum severity of log messages going to the memory buffer of recent log events. Messages with a lower severity than the specified value will not be cached in the buffer. More Info...

Filter: None — The filter configuration for messages that are stored in the log memory buffer. By default, all log messages are cached. More Info...

Message Catalog

- Message catalogs are available in HTML format on e-docs as part of the documentation deliverable. You can search for messages by error number using the search engine.



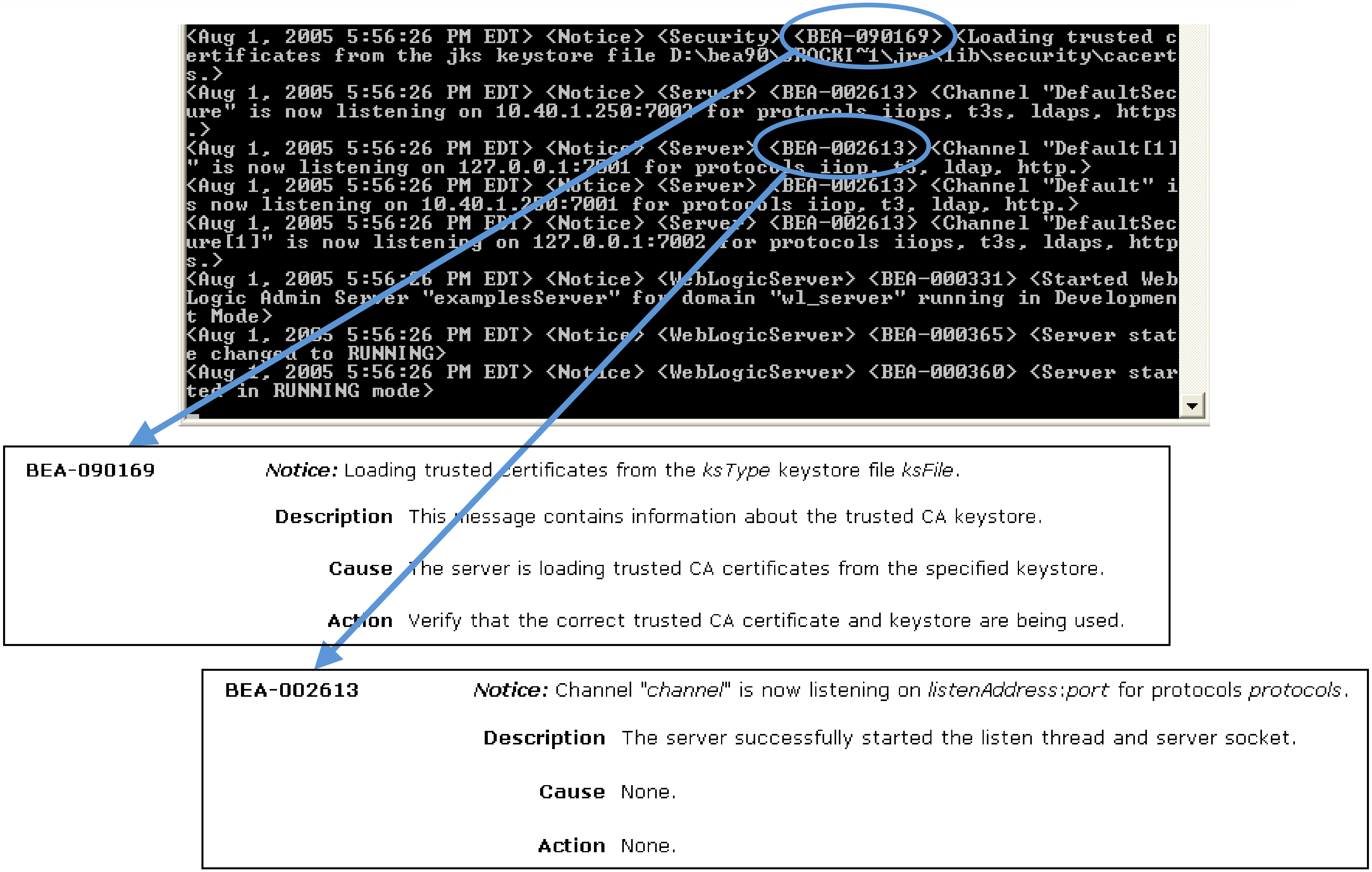
eDocs Home > BEA WebLogic Server and WebLogic Express 9.0 Documentation > Index of Messages by Message Range (by Subsystem)

Index of Messages by Message Range

Messages in the Message Catalog are part of the WebLogic Server Internationalization and Localization packages.

Range	Subsystem	Catalog
BEA-000100 - BEA-000199	Cluster	Cluster
BEA-000200 - BEA-000399	WebLogicServer	T3Srvr
BEA-000400 - BEA-000499	Socket	Socket
000500 - 000599	RJVM	RJVM
BEA-000600 - BEA-000699	Common	Common
BEA-000700 - BEA-000799	T3Misc	T3Misc

Message Catalog



```
<Aug 1, 2005 5:56:26 PM EDT> <Notice> <Security> <BEA-090169> <Loading trusted c
ertificates from the jks keystore file D:\bea90\JROCKI~1\jre\lib\security\cacert
s.>
<Aug 1, 2005 5:56:26 PM EDT> <Notice> <Server> <BEA-002613> <Channel "DefaultSec
ure" is now listening on 10.40.1.250:7002 for protocols iiops, t3s, ldaps, https
.>
<Aug 1, 2005 5:56:26 PM EDT> <Notice> <Server> <BEA-002613> <Channel "Default[1]
" is now listening on 127.0.0.1:7001 for protocols iiop, t3, ldap, http.>
<Aug 1, 2005 5:56:26 PM EDT> <Notice> <Server> <BEA-002613> <Channel "Default" i
s now listening on 10.40.1.250:7001 for protocols iiop, t3, ldap, http.>
<Aug 1, 2005 5:56:26 PM EDT> <Notice> <Server> <BEA-002613> <Channel "DefaultSec
ure[1]" is now listening on 127.0.0.1:7002 for protocols iiops, t3s, ldaps, http
s.>
<Aug 1, 2005 5:56:26 PM EDT> <Notice> <WebLogicServer> <BEA-000331> <Started Web
Logic Admin Server "examplesServer" for domain "wl_server" running in Developmen
t Mode>
<Aug 1, 2005 5:56:26 PM EDT> <Notice> <WebLogicServer> <BEA-000365> <Server stat
e changed to RUNNING>
<Aug 1, 2005 5:56:26 PM EDT> <Notice> <WebLogicServer> <BEA-000360> <Server star
ted in RUNNING mode>
```

BEA-090169 *Notice:* Loading trusted certificates from the *ksType* keystore file *ksFile*.

Description This message contains information about the trusted CA keystore.

Cause The server is loading trusted CA certificates from the specified keystore.

Action Verify that the correct trusted CA certificate and keystore are being used.

BEA-002613 *Notice:* Channel "*channel*" is now listening on *listenAddress*:*port* for protocols *protocols*.

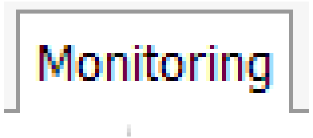
Description The server successfully started the listen thread and server socket.

Cause None.

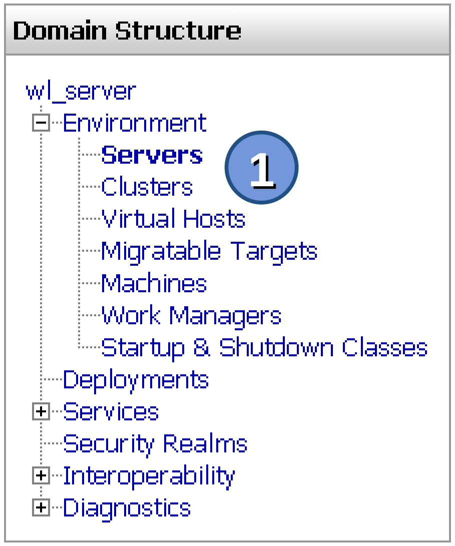
Action None.

Using the Console to Monitor

- The Administration Console offers some monitoring capabilities:

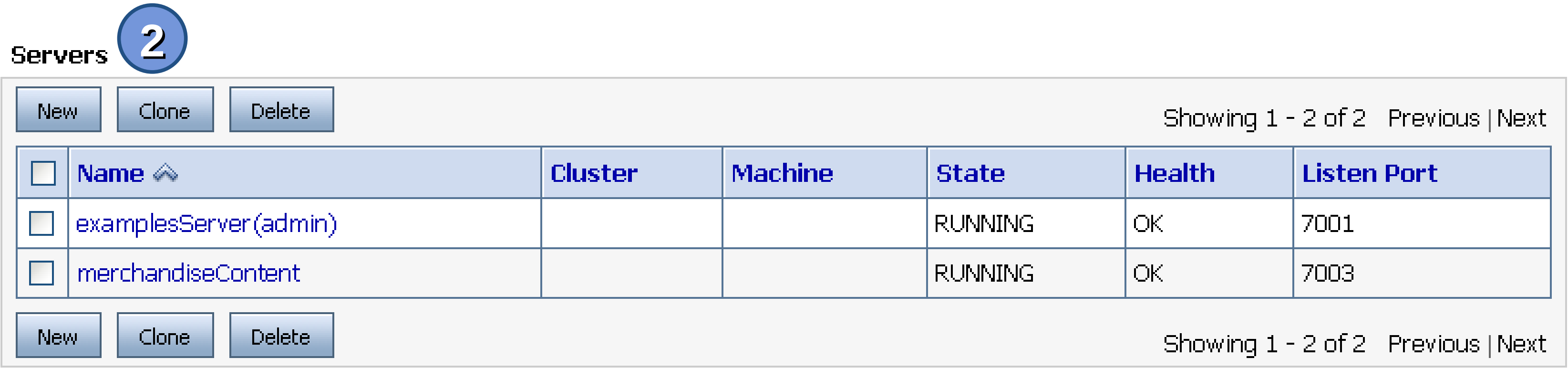
Attribute	Description
 Monitoring	Many of the Console's objects have a Monitoring tab, that allows you to view monitoring information for that object
Customize this table	The monitoring view can be customized by clicking on Customize this table…

Monitoring Running Servers



Domain Structure

- wl_server
  - ⊟ Environment
    - **Servers**
    - Clusters
    - Virtual Hosts
    - Migratable Targets
    - Machines
    - Work Managers
    - Startup & Shutdown Classes
  - Deployments
  - ⊞ Services
  - Security Realms
  - ⊞ Interoperability
  - ⊞ Diagnostics



Servers

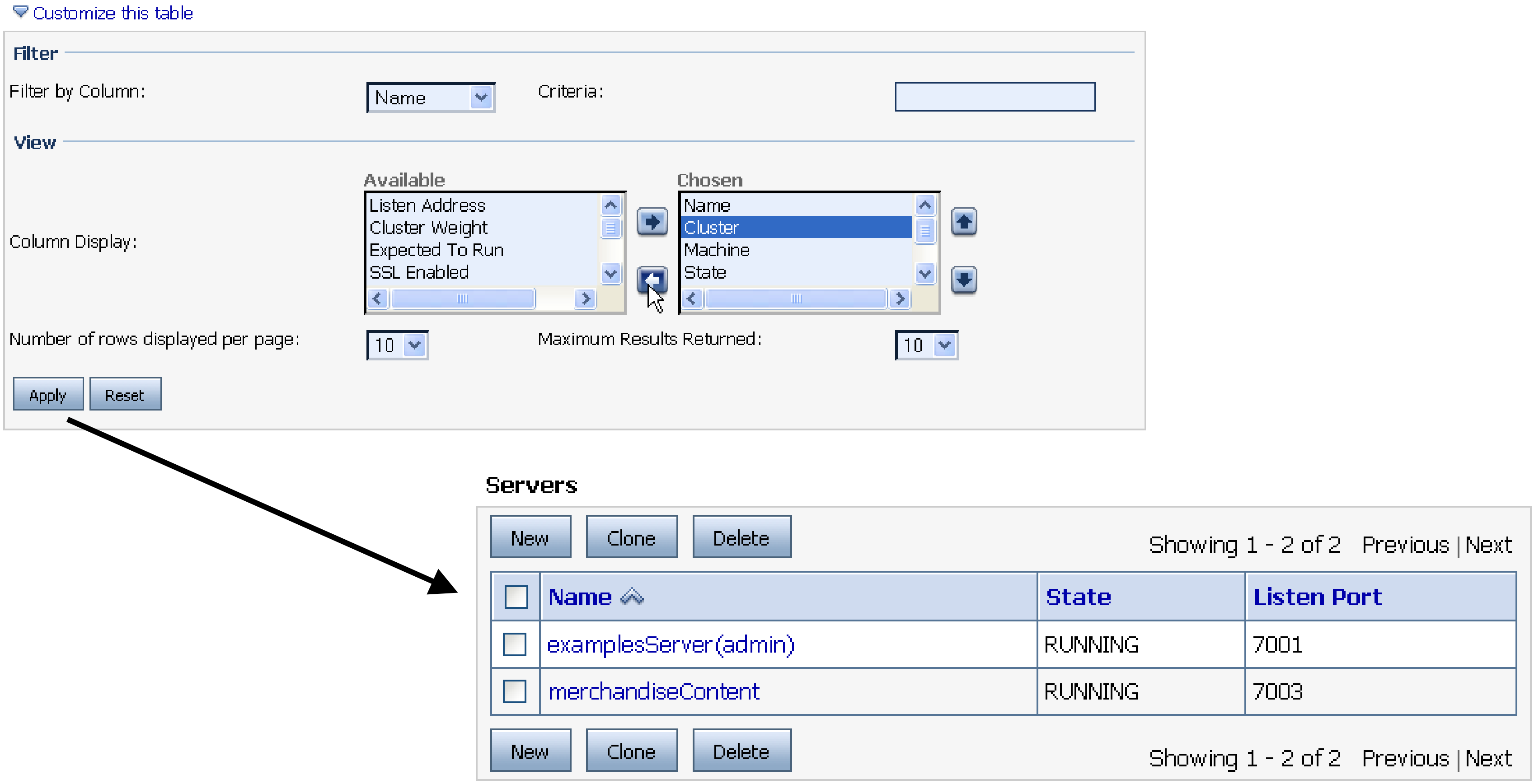
Showing 1 - 2 of 2 Previous | Next

□	Name	Cluster	Machine	State	Health	Listen Port
□	examplesServer(admin)			RUNNING	OK	7001
□	merchandiseContent			RUNNING	OK	7003

Showing 1 - 2 of 2 Previous | Next

Customizing views

- Columns can be customized on views



Filter

Filter by Column: Name

Criteria:

View

Column Display:

Available: Listen Address, Cluster Weight, Expected To Run, SSL Enabled

Chosen: Name, Cluster, Machine, State

Number of rows displayed per page: 10

Maximum Results Returned: 10

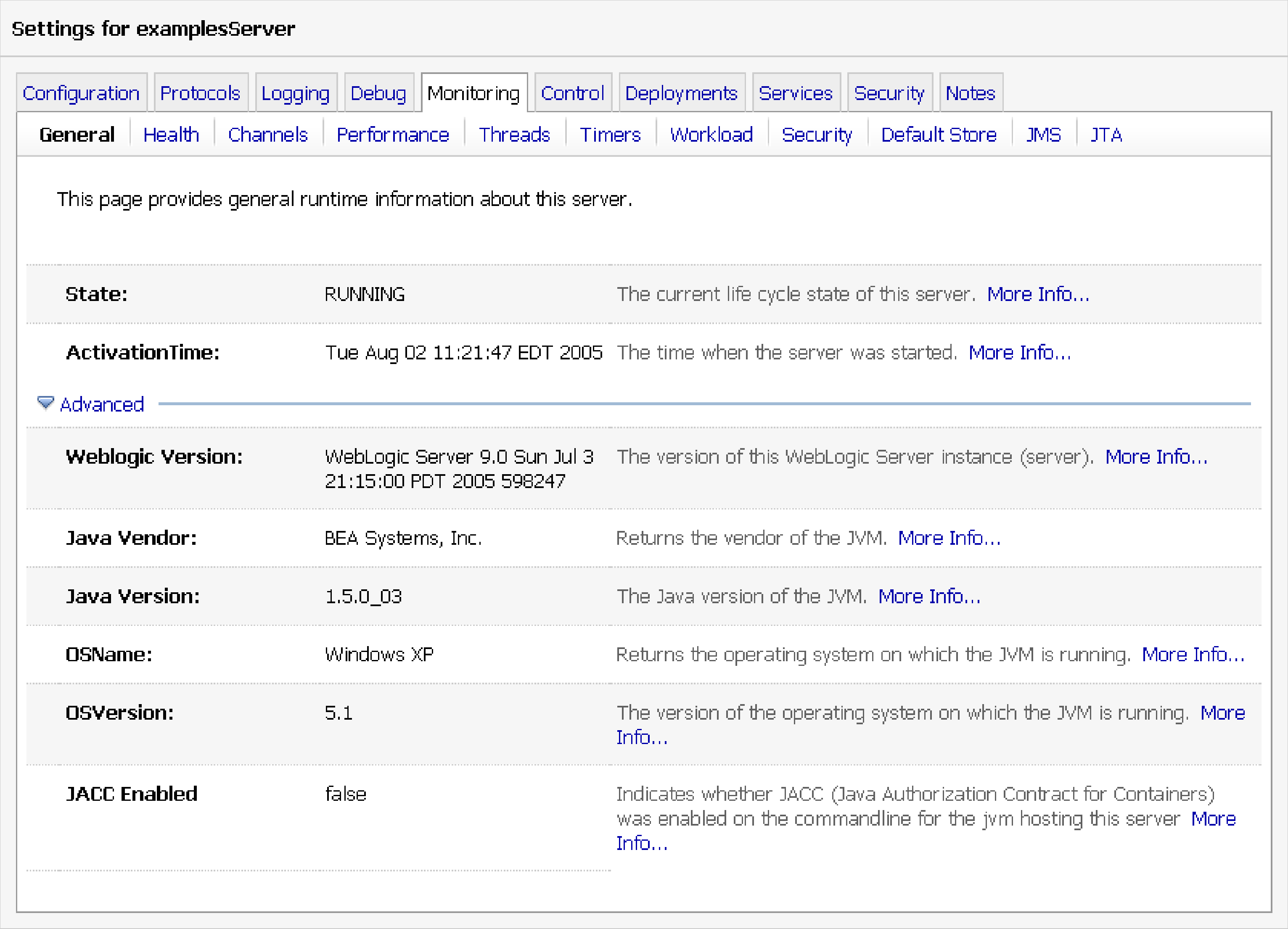
Servers

Showing 1 - 2 of 2 Previous | Next

□	Name	State	Listen Port
□	examplesServer(admin)	RUNNING	7001
□	merchandiseContent	RUNNING	7003

Showing 1 - 2 of 2 Previous | Next

Monitoring Individual Servers



Settings for examplesServer

This page provides general runtime information about this server.

State:	RUNNING	The current life cycle state of this server. More Info...
ActivationTime:	Tue Aug 02 11:21:47 EDT 2005	The time when the server was started. More Info...

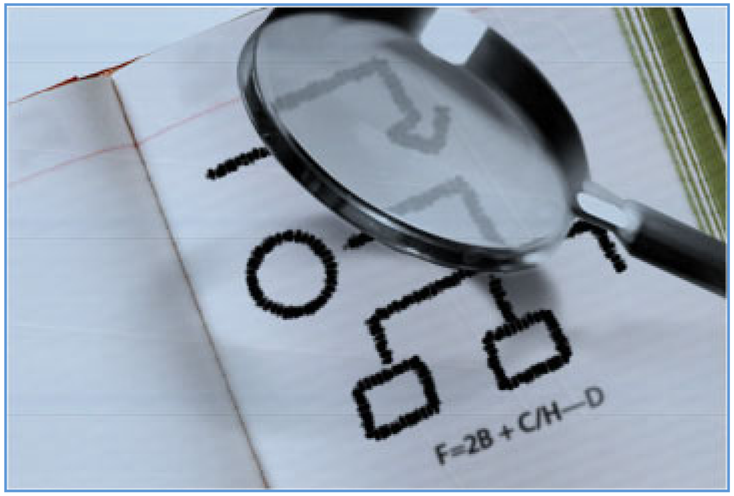
Advanced

Weblogic Version:	WebLogic Server 9.0 Sun Jul 3 21:15:00 PDT 2005 598247	The version of this WebLogic Server instance (server). More Info...
Java Vendor:	BEA Systems, Inc.	Returns the vendor of the JVM. More Info...
Java Version:	1.5.0_03	The Java version of the JVM. More Info...
OSName:	Windows XP	Returns the operating system on which the JVM is running. More Info...
OSVersion:	5.1	The version of the operating system on which the JVM is running. More Info...
JACC Enabled	false	Indicates whether JACC (Java Authorization Contract for Containers) was enabled on the commandline for the jvm hosting this server More Info...

Section Review

In this section we discussed:

- Using Log Files
- Monitoring Servers



Road Map

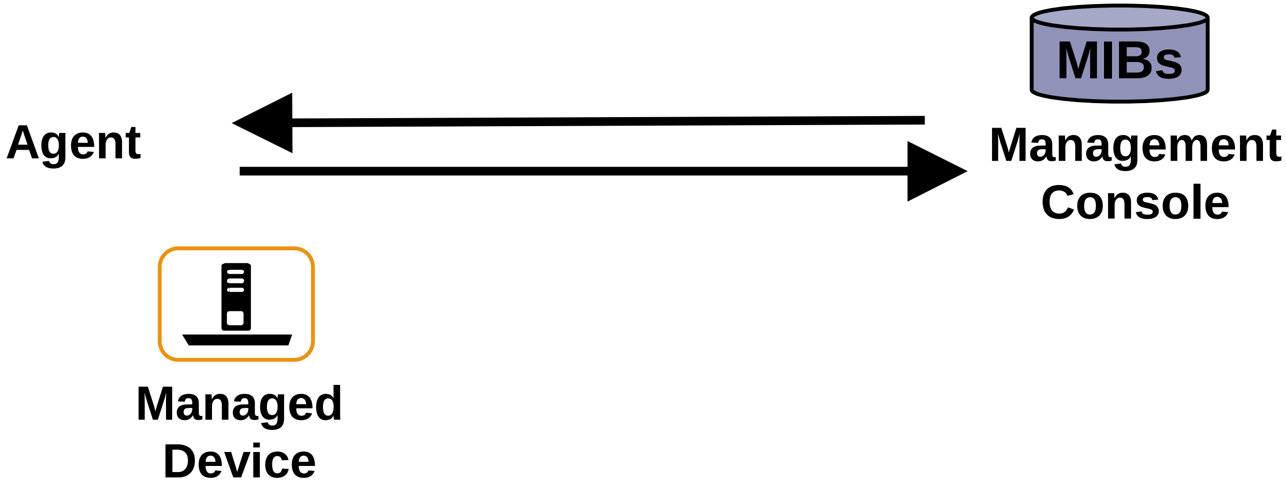
1. Remote Administration
2. Logs and Monitoring
3. **SNMP Concepts**
   – Architecture, MIB, OID
   – SNMP Agent
   – Trap Notifications
   – SNMP Features of WLS
4. WLS SNMP Agent
5. WLS SNMP Management Tools
6. Network Channels

SNMP

- The Simple Network Management Protocol (SNMP) is a protocol for managing distributed devices.
- Examples of devices include:
  - bridges
  - routers
  - servers
  - printers

SNMP Architecture

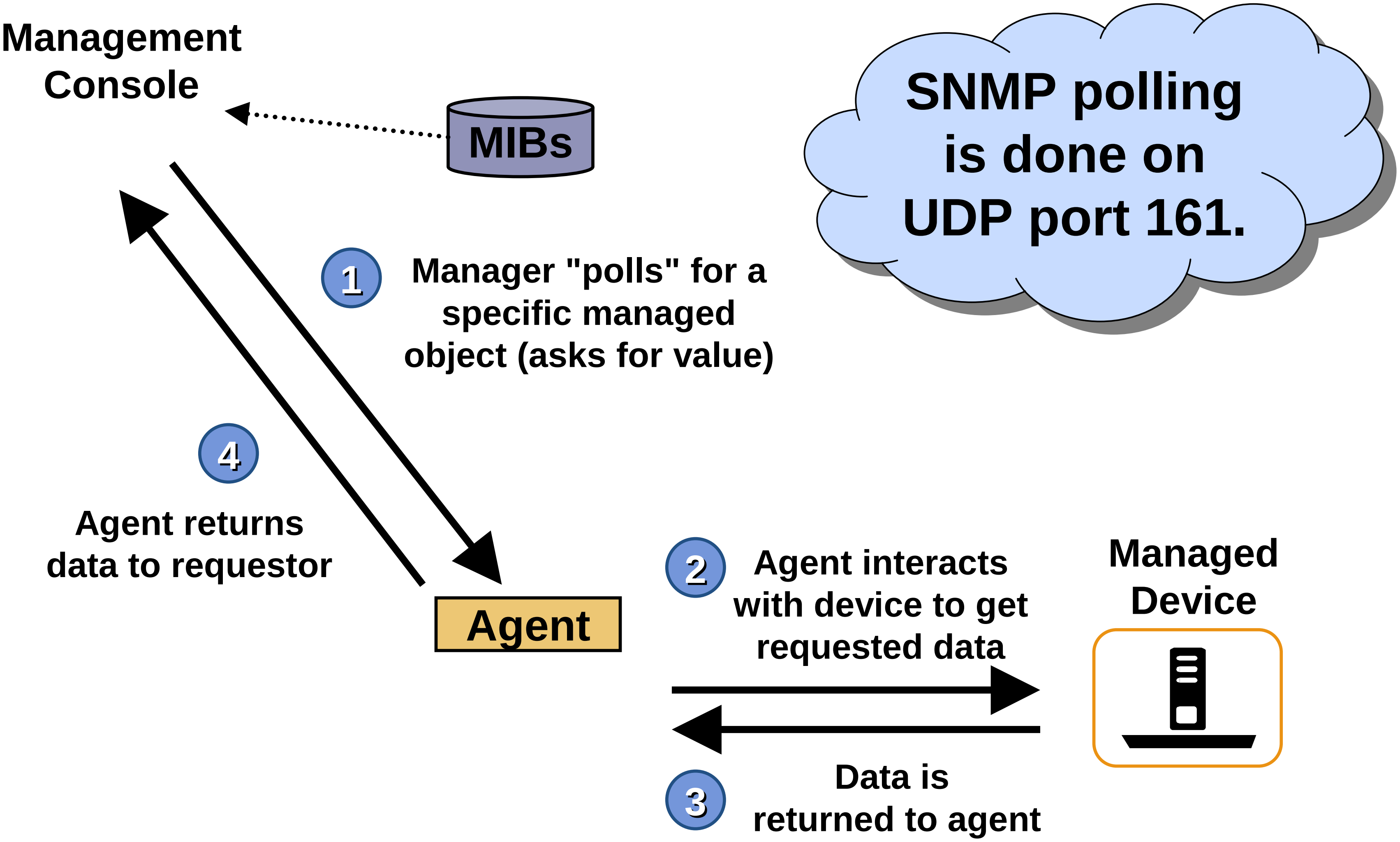
- SNMP works by monitoring devices through software known as *agents*.
- Agents report information to a *manager*:
  - on demand (*polling*)
  - automatically (*traps*)



Management Information Base (MIB)

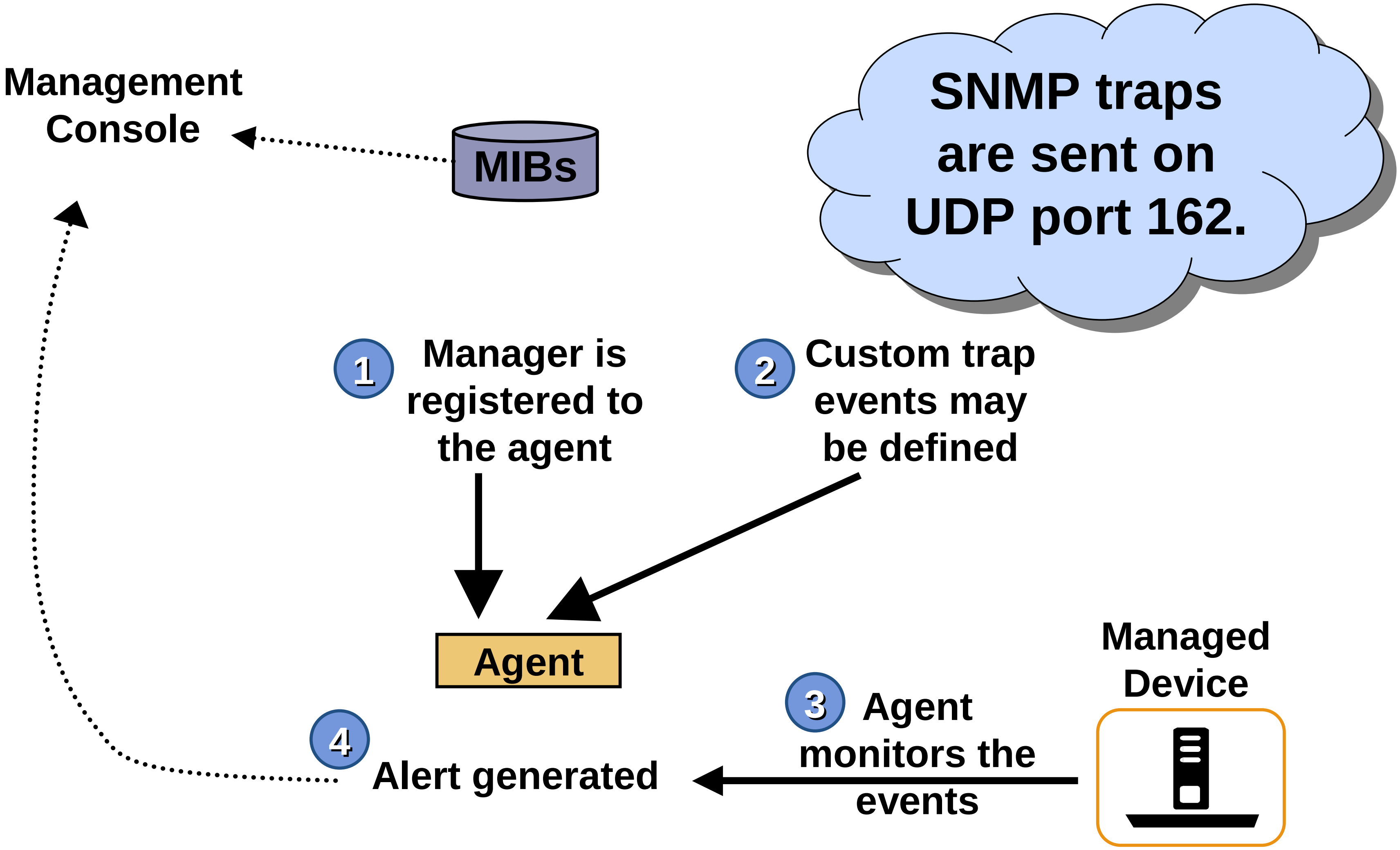
- A “managed object” is a value that can be monitored by an Agent.
- A Management Information Base (MIB) is a file that:
  - contains a list of these objects
  - is related to a single device type
  - is used by the manager to:
    - determine the available objects that can be polled, and
    - make sense of values returned by trap notifications

SNMP Polling


Management Console
MIBs
1
Manager "polls" for a specific managed object (asks for value)
4
Agent returns data to requestor
Agent
2
Agent interacts with device to get requested data
Managed Device
3
Data is returned to agent

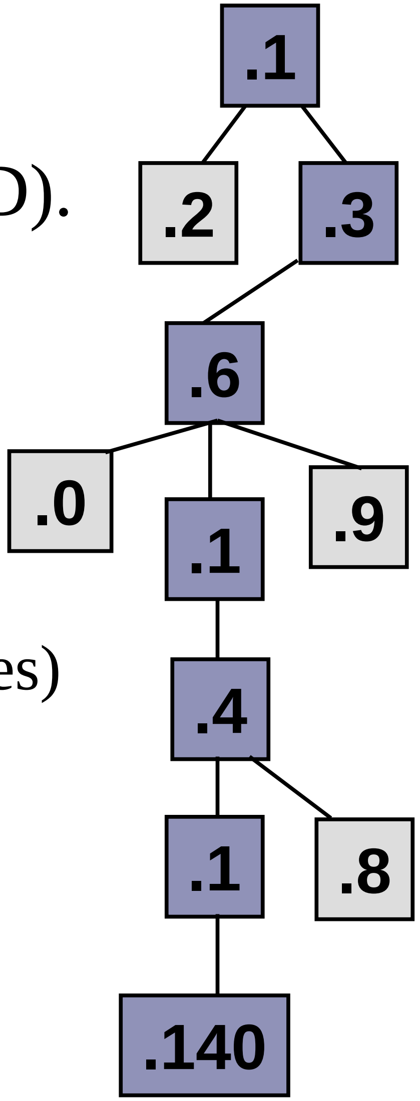
SNMP polling is done on UDP port 161.

SNMP Traps


Management Console
MIBs
SNMP traps are sent on UDP port 162.
1 Manager is registered to the agent
2 Custom trap events may be defined
Agent
Managed Device
3 Agent monitors the events
4 Alert generated

OIDs

- Each managed object is represented by an identifier, called the Object IDentifier (OID).
- The OIDs:
  - are represented as dot-separated integers (e.g.: `.1.3.6.1.4.1.140` …)
  - are hierarchical
  - refer to single objects (leaf) or groups (branches)



The Root for WLS OIDs

- The base for all objects in WLS is:

  `.1.3.6.1.4.1.140.625`

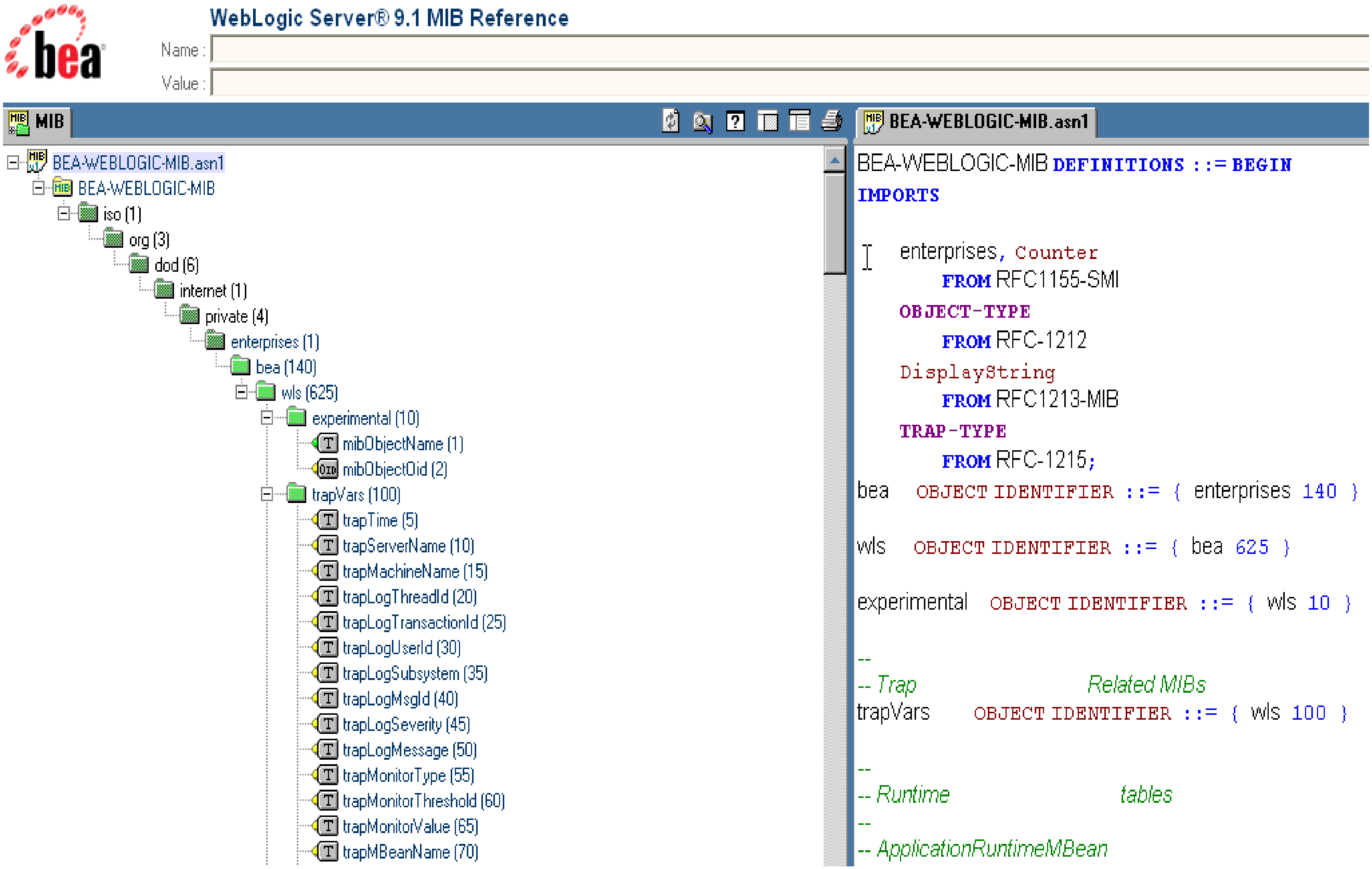
- All WLS SNMP objects are located on some hierarchical level under the root, e.g.:

  To know the current operating system, you can query the managed object `jvmRuntimeOSName`, located under the OID `.1.3.6.1.4.1.140.625.340.1.45`

WebLogic Server 9.1 MIB Reference

- The available managed objects and their OIDs can be looked up online :
  - `http://e-docs.bea.com/wls/docs91/snmp`
- locate the OID root for an object and write it down
- your SNMP manager tool can then use this OID root to poll objects under it

WebLogic Server 9.1 MIB Reference



WebLogic Server® 9.1 MIB Reference

Name :

Value :

```
BEA-WEBLOGIC-MIB.asn1
  BEA-WEBLOGIC-MIB
    iso (1)
      org (3)
        dod (6)
          internet (1)
            private (4)
              enterprises (1)
                bea (140)
                  wls (625)
                    experimental (10)
                      mibObjectName (1)
                      mibObjectOid (2)
                    trapVars (100)
                      trapTime (5)
                      trapServerName (10)
                      trapMachineName (15)
                      trapLogThreadId (20)
                      trapLogTransactionId (25)
                      trapLogUserId (30)
                      trapLogSubsystem (35)
                      trapLogMsgId (40)
                      trapLogSeverity (45)
                      trapLogMessage (50)
                      trapMonitorType (55)
                      trapMonitorThreshold (60)
                      trapMonitorValue (65)
                      trapMBeanName (70)
```

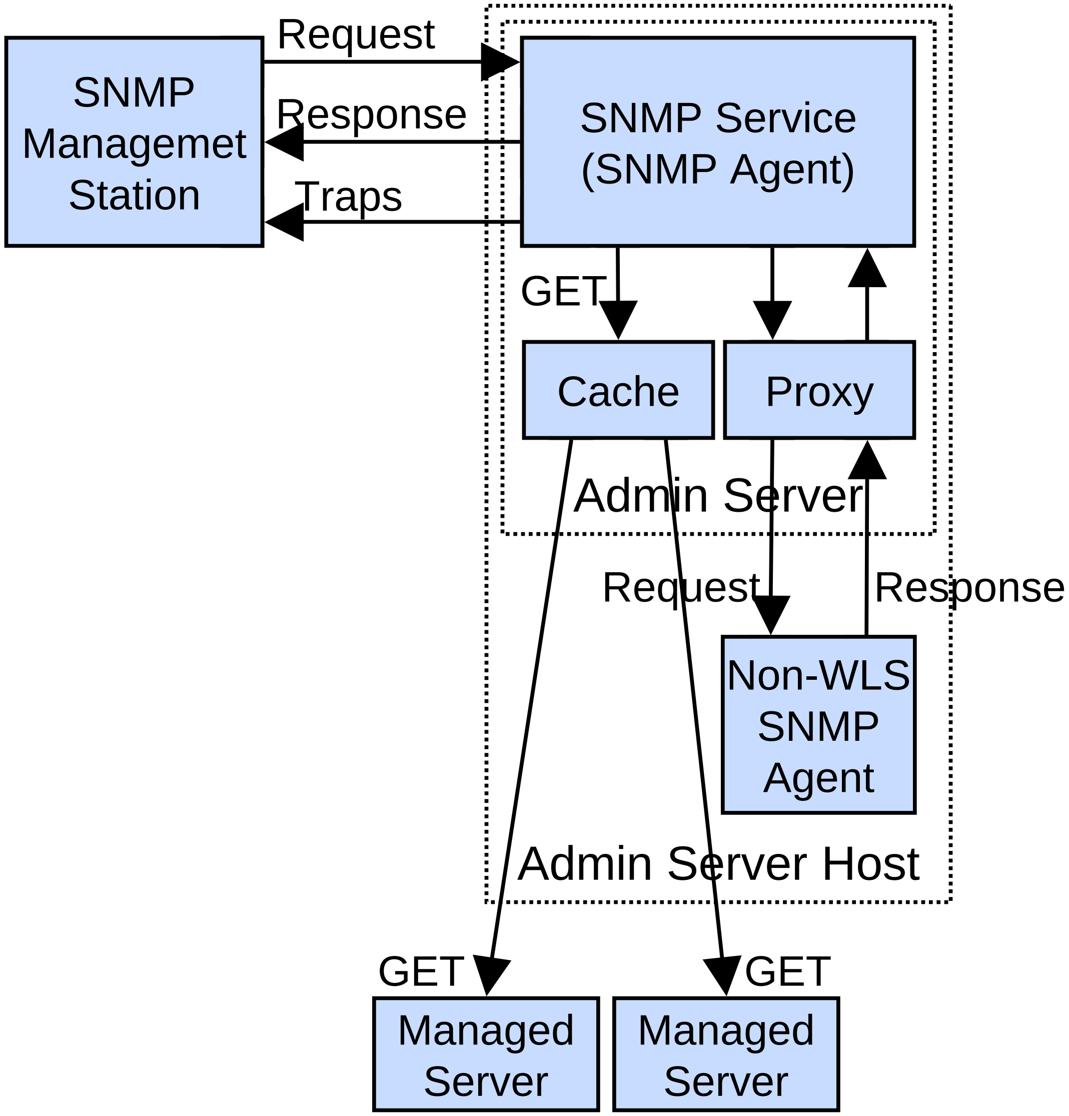
```
BEA-WEBLOGIC-MIB DEFINITIONS ::= BEGIN
IMPORTS

    enterprises, Counter
        FROM RFC1155-SMI
    OBJECT-TYPE
        FROM RFC-1212
    DisplayString
        FROM RFC1213-MIB
    TRAP-TYPE
        FROM RFC-1215;
bea   OBJECT IDENTIFIER ::= { enterprises 140 }

wls   OBJECT IDENTIFIER ::= { bea 625 }

experimental   OBJECT IDENTIFIER ::= { wls 10 }

--
-- Trap                        Related MIBs
trapVars     OBJECT IDENTIFIER ::= { wls 100 }

--
-- Runtime                        tables
--
-- ApplicationRuntimeMBean
```

WLS SNMP Support

- WLS provides an SNMP Agent that:
  - provides monitoring capability to SNMP managers
  - generates standard and user-defined trap notification sent to registered managers
  - runs inside the administration server (`weblogic.Server`)
  - doesn’t support the SET operation

WLS SNMP Architecture

- The WLS SNMP Agent:
  - caches its data and refreshes the cache regularly
  - has the ability to proxy other SNMP agents



WLS Managed Objects

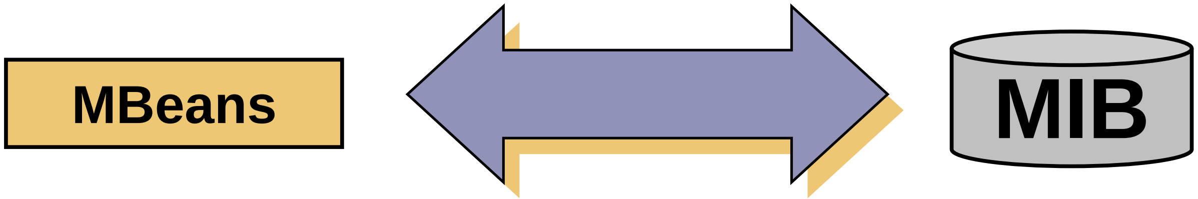
- The WLS MIB supports polling for hundreds of managed objects, e.g.:
  - domain, Web server, clustering
  - deployment
  - applications (enterprise, EJB, Web)
  - execute queues
  - JDBC, JMS, JTA services
  - JVM information

WLS Traps

- The WLS MIB defines standard trapping notifications for:
  - server start
  - server shutdown
  - MBean attribute changed
  - logging notification
  - MBean monitoring notification (gauge, string, counter)
- The last three allow user-defined trap notifications monitored by the agent.

SNMP and WLS MBeans

- In WLS, SNMP and MBeans are closely related because:
  - internally, managed objects map to MBean attributes
  - user-defined traps test MBean attributes for certain conditions



WLS SNMP Management Tools

- WLS comes with command-line management utilities that can:
  - poll information (one managed object, or all of them under a branch)
  - alert the user of all trap notifications
  - generate trap events for testing

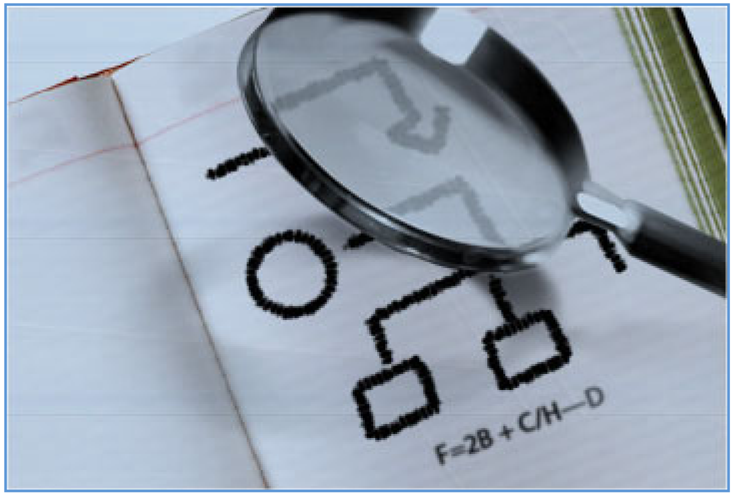
SNMP Vendors

- Some SNMP management systems compatible with WLS 9.1 include:
  - IBM Tivoli
  - HP Openview
  - Sun Domain/SunNet/Site Manager
  - CA Unicenter

Section Review

In this section we discussed:

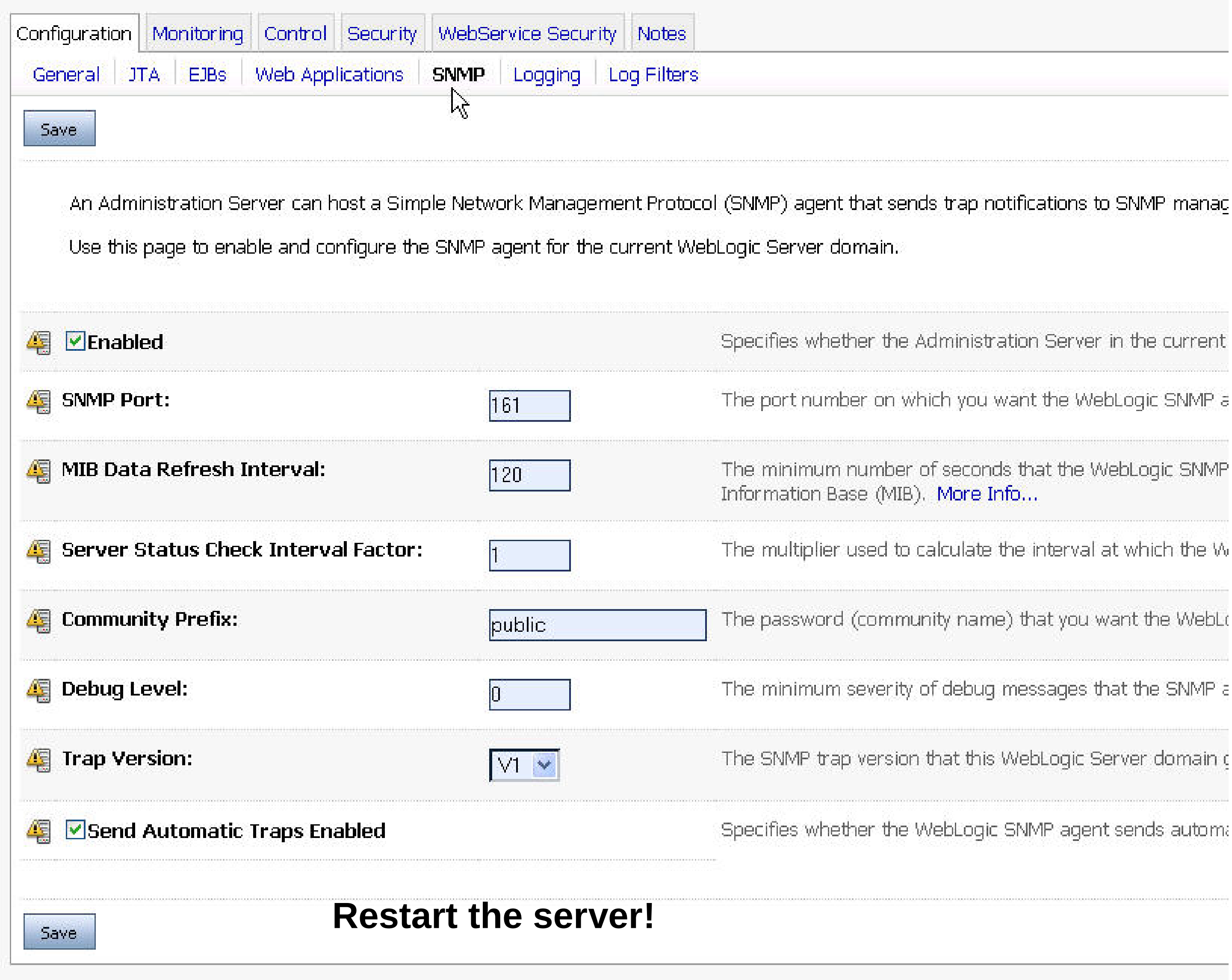
- ✓ SNMP definitions:
  - Agent
  - Manager
  - managed object
  - MIB
  - OID
  - polling
  - traps
- ✓ WLS support for SNMP



Road Map

1. Remote Administration
2. Logs and Monitoring
3. SNMP Concepts
4. **WLS SNMP Agent**
   - Activating the SNMP Agent
   - Registering Managers to Receive Traps
   - Setting Up Traps
5. WLS SNMP Management Tools
6. Network Channels

Turning On the WLS SNMP Agent



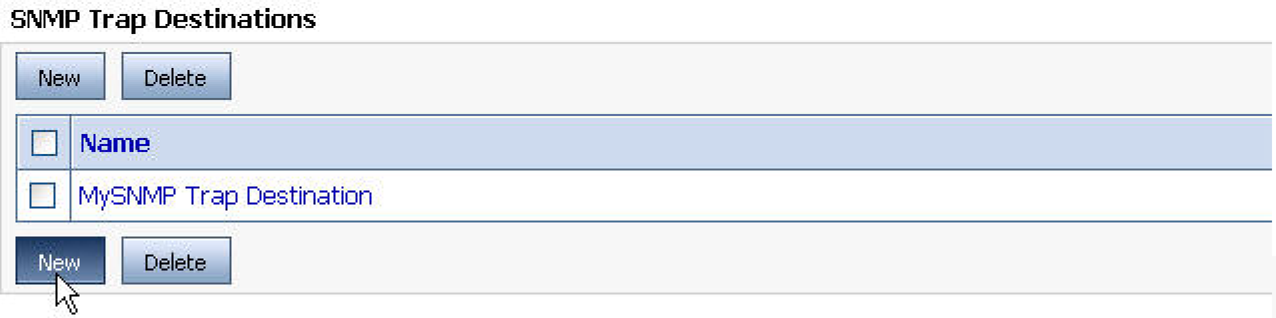
An Administration Server can host a Simple Network Management Protocol (SNMP) agent that sends trap notifications to SNMP manag

Use this page to enable and configure the SNMP agent for the current WebLogic Server domain.

☑ Enabled		Specifies whether the Administration Server in the current
SNMP Port:	161	The port number on which you want the WebLogic SNMP a
MIB Data Refresh Interval:	120	The minimum number of seconds that the WebLogic SNMP Information Base (MIB). More Info...
Server Status Check Interval Factor:	1	The multiplier used to calculate the interval at which the W
Community Prefix:	public	The password (community name) that you want the WebL
Debug Level:	0	The minimum severity of debug messages that the SNMP a
Trap Version:	V1	The SNMP trap version that this WebLogic Server domain g
☑ Send Automatic Traps Enabled		Specifies whether the WebLogic SNMP agent sends automa

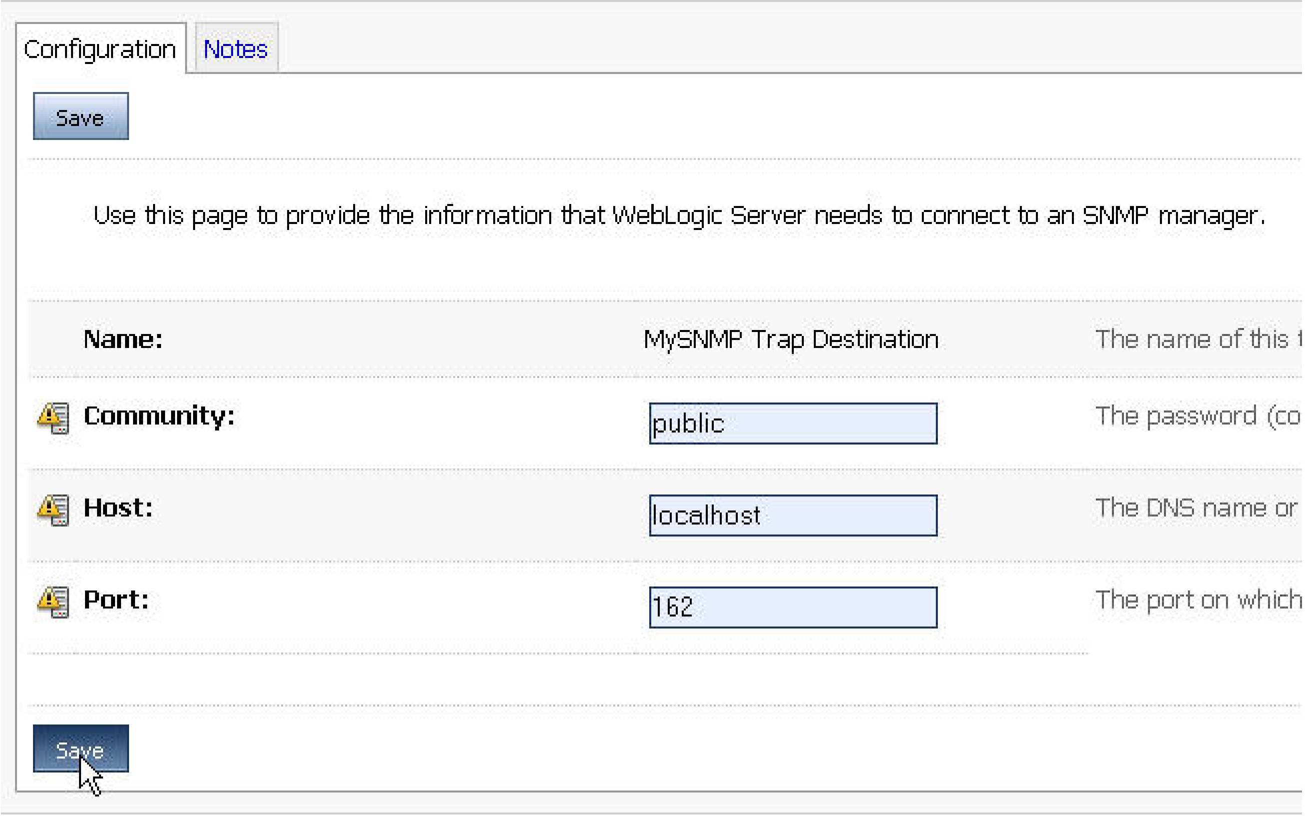
Restart the server!

Registering Managers for Traps



SNMP Trap Destinations

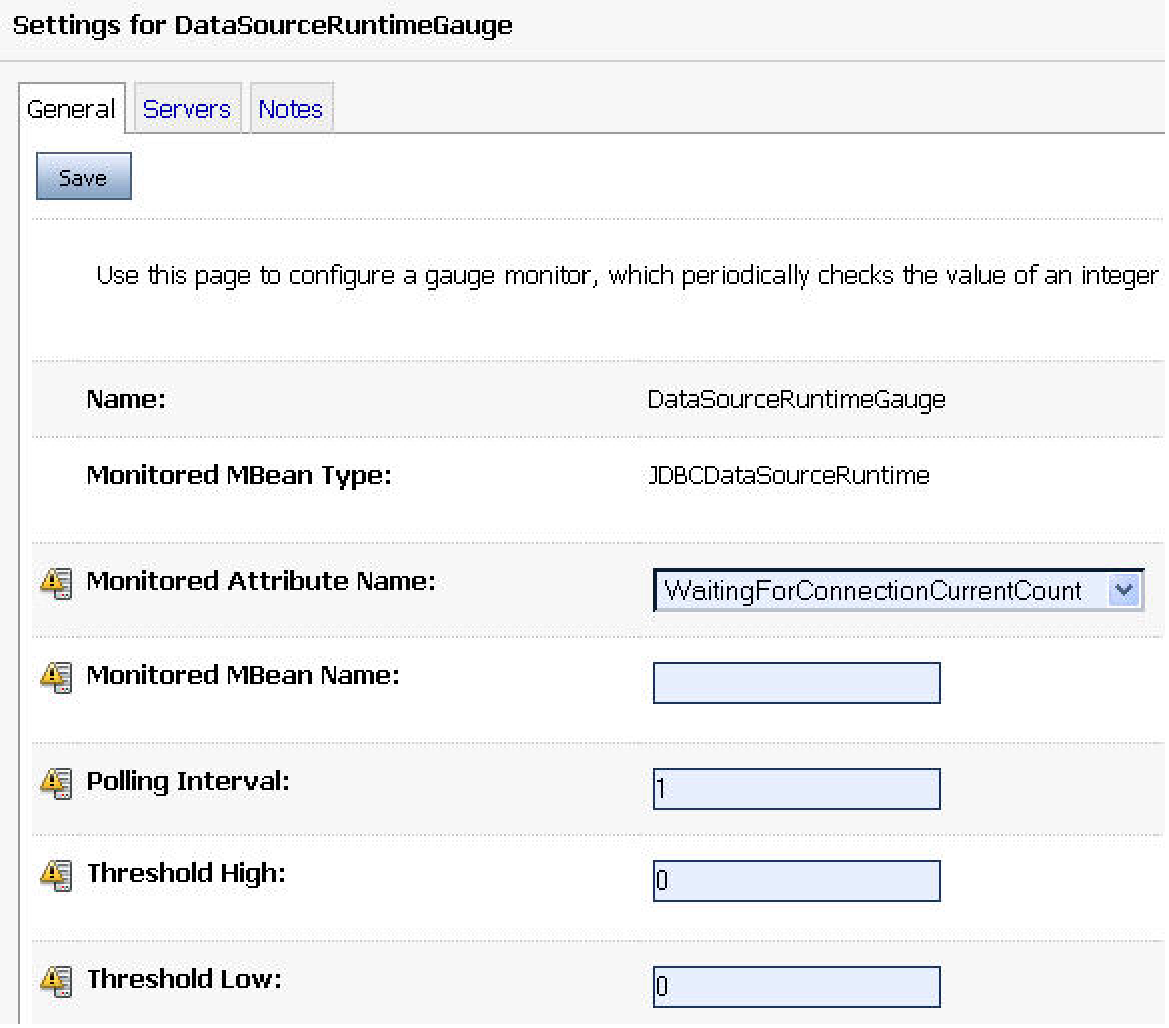
□	Name
□	MySNMP Trap Destination



Use this page to provide the information that WebLogic Server needs to connect to an SNMP manager.

Name:	MySNMP Trap Destination	The name of this t
Community:	public	The password (co
Host:	localhost	The DNS name or
Port:	162	The port on which

Creating User-Defined Traps



Settings for DataSourceRuntimeGauge

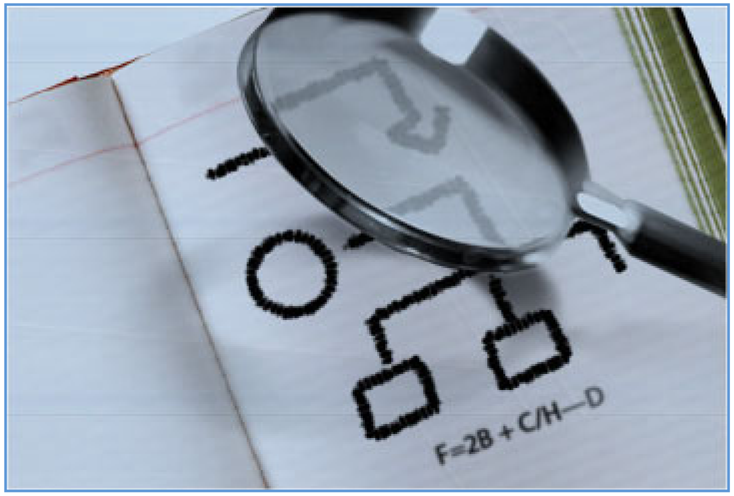
Use this page to configure a gauge monitor, which periodically checks the value of an integer

Name:	DataSourceRuntimeGauge
Monitored MBean Type:	JDBCDataSourceRuntime
Monitored Attribute Name:	WaitingForConnectionCurrentCount
Monitored MBean Name:	
Polling Interval:	1
Threshold High:	0
Threshold Low:	0

Section Review

In this section we discussed:

- ✓ Configuring the WLS SNMP Agent
- ✓ Registering managers to receive traps
- ✓ Setting up custom traps



Road Map

1. Remote Administration
2. Logs and Monitoring
3. SNMP Concepts
4. WLS SNMP Agent
5. **WLS SNMP Management Tools**
   - Overview
   - Using `snmpwalk` and `snmptrapd`
6. Network Channels

SNMP Tools

- WebLogic Server supports five testing tools for testing SNMP:
  - `snmpwalk:` return all data using SNMP `GET` and `GETNEXT` request for tabular data.
  - `snmptrapd:` receive and dump SNMP traps.
  - `snmpv1trap:` generate a test SNMP trap.
  - `snmpget:` return information from an agent using SNMP `GET`.
  - `snmpgetnext:` return information using SNMP `GETNEXT`.

Getting All Objects In a Branch

- `snmpwalk` traverses all managed objects in a branch and writes them out.

Syntax:

```
java snmpwalk [-p <port>] [-c <community>] <host> <OID>
```

Arguments:

port The port for the trap notifications; see agent's configuration. The default is 161.

community The password-like identifier which this manager tool will use. The default is 'public'.

host The address of the agent to poll.

OID The full numeric object identifier of the branch to traverse.

Listening to Trap Notifications

- `snmptrapd` listens to trap notifications from an agent, and displays them.

Syntax:

```
java snmptrapd [-p <port>] [-c <community>]
```

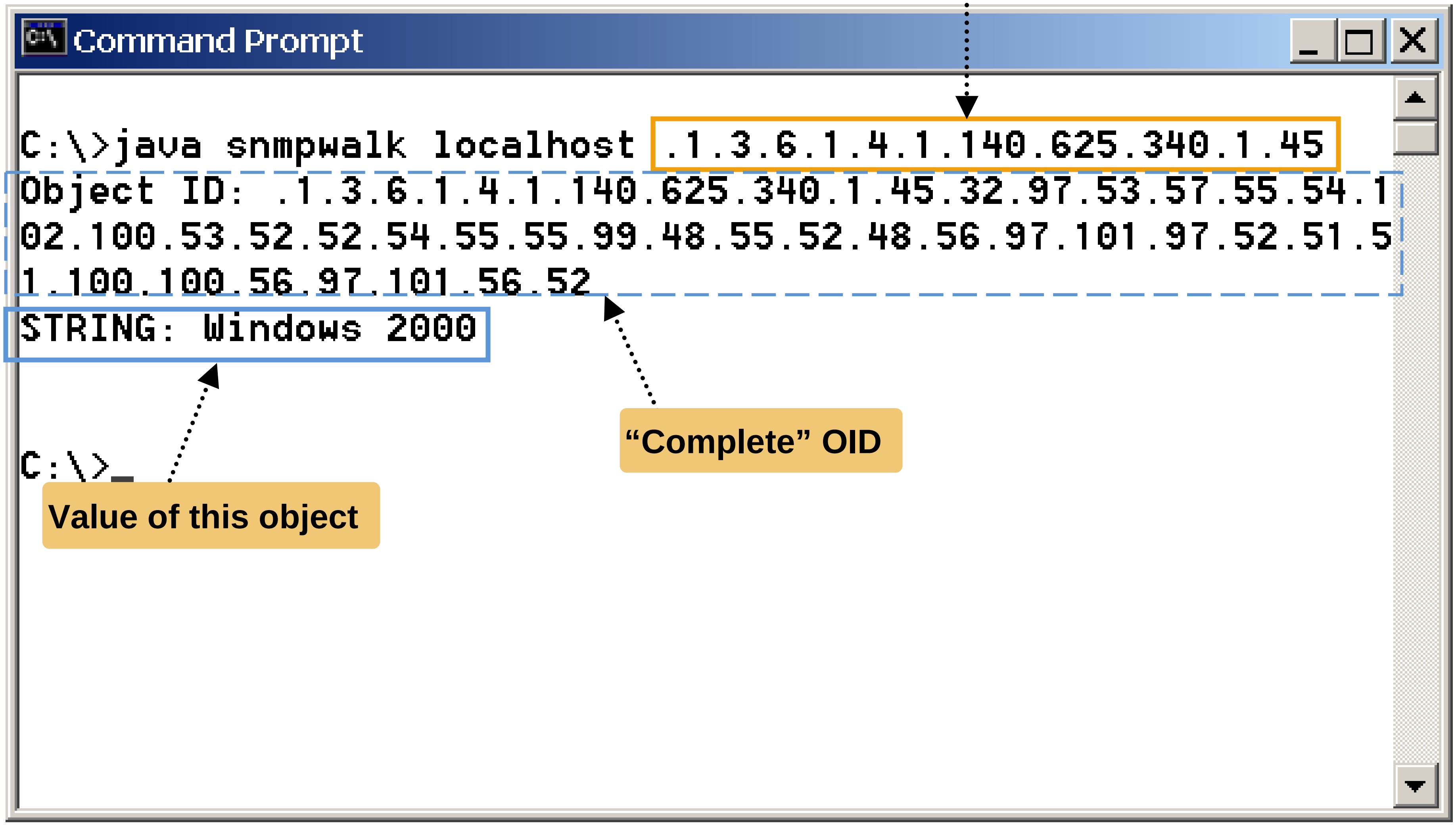
Arguments:

`port` The port for the trap notifications; see agent’s configuration. The default is 162.

`community` The password-like identifier which this manager tool will use. The default is ‘public’.

Example: Polling an Object

OID root for `jvmRuntimeOSName`

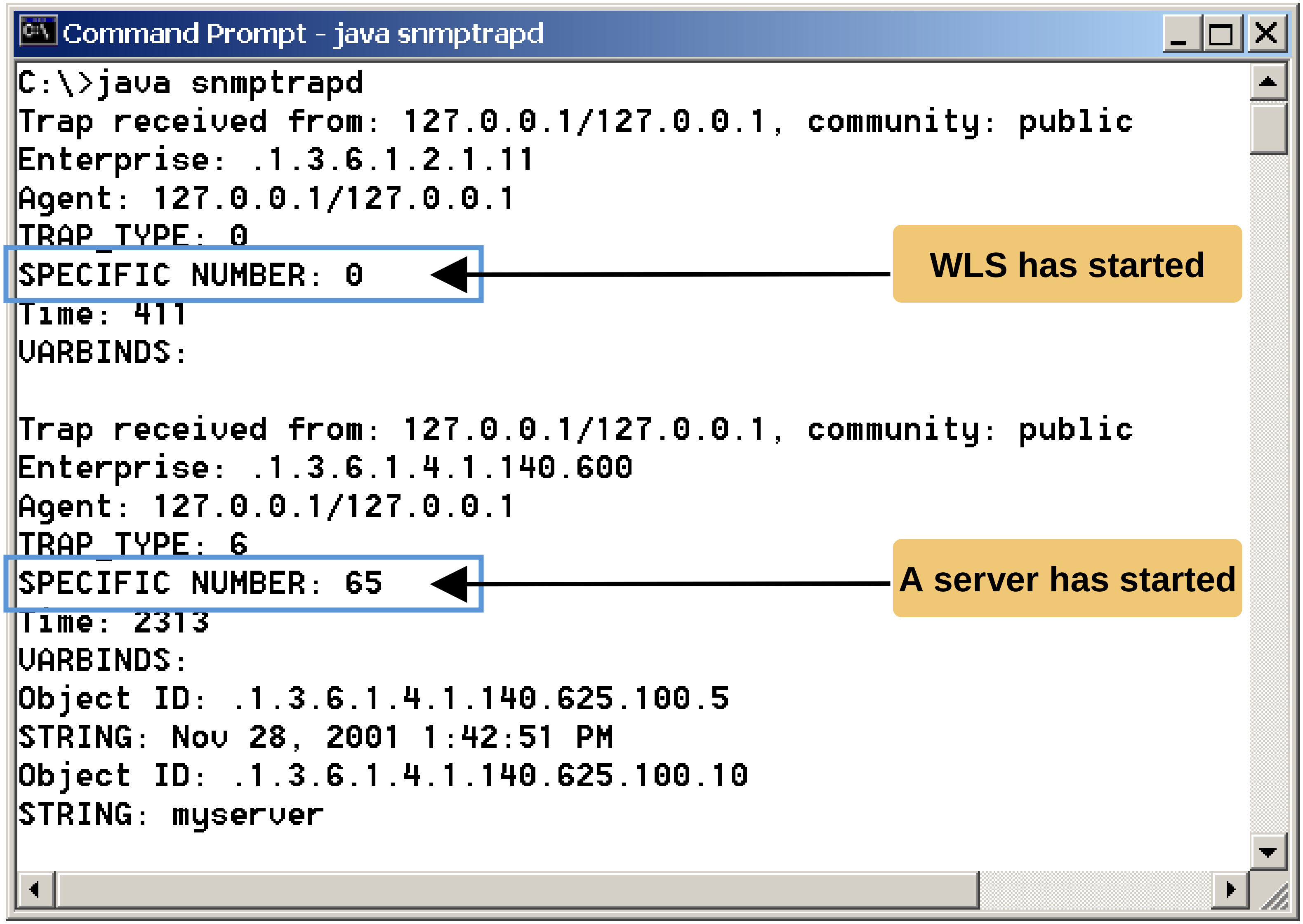


```
C:\>java snmpwalk localhost .1.3.6.1.4.1.140.625.340.1.45
Object ID: .1.3.6.1.4.1.140.625.340.1.45.32.97.53.57.55.54.1
02.100.53.52.52.54.55.55.99.48.55.52.48.56.97.101.97.52.51.5
1.100.100.56.97.101.56.52
STRING: Windows 2000


C:\>_
```

“Complete” OID

Value of this object

Example: Catching a Trap



```
C:\>java snmptrapd
Trap received from: 127.0.0.1/127.0.0.1, community: public
Enterprise: .1.3.6.1.2.1.11
Agent: 127.0.0.1/127.0.0.1
TRAP_TYPE: 0
SPECIFIC NUMBER: 0
Time: 411
VARBINDS:

Trap received from: 127.0.0.1/127.0.0.1, community: public
Enterprise: .1.3.6.1.4.1.140.600
Agent: 127.0.0.1/127.0.0.1
TRAP_TYPE: 6
SPECIFIC NUMBER: 65
Time: 2313
VARBINDS:
Object ID: .1.3.6.1.4.1.140.625.100.5
STRING: Nov 28, 2001 1:42:51 PM
Object ID: .1.3.6.1.4.1.140.625.100.10
STRING: myserver
```

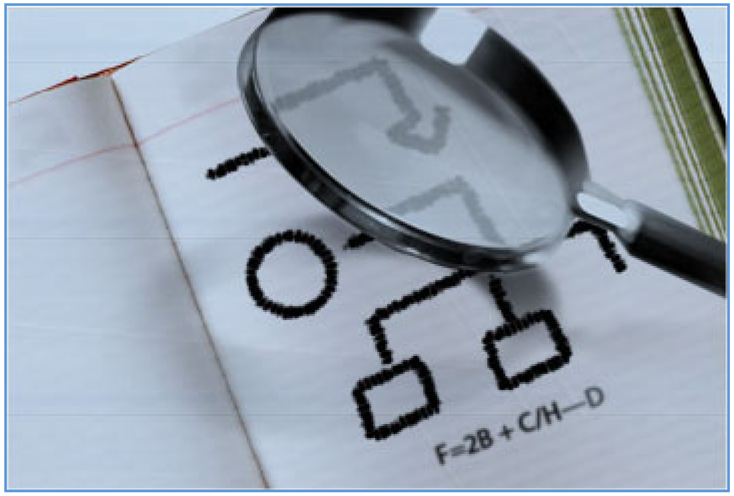
WLS has started

A server has started

Section Review

In this section we discussed:

- ✓ WLS-provided SNMP management tools
- ✓ Using `snmpwalk` and `snmptrapd`



Exercise

Using SNMP with WebLogic Server

- In this lab you will use Node Manager to control managed servers.
- Ask the instructor for any clarification.
- The instructor will determine the stop time.

Lab Exercise

Road Map

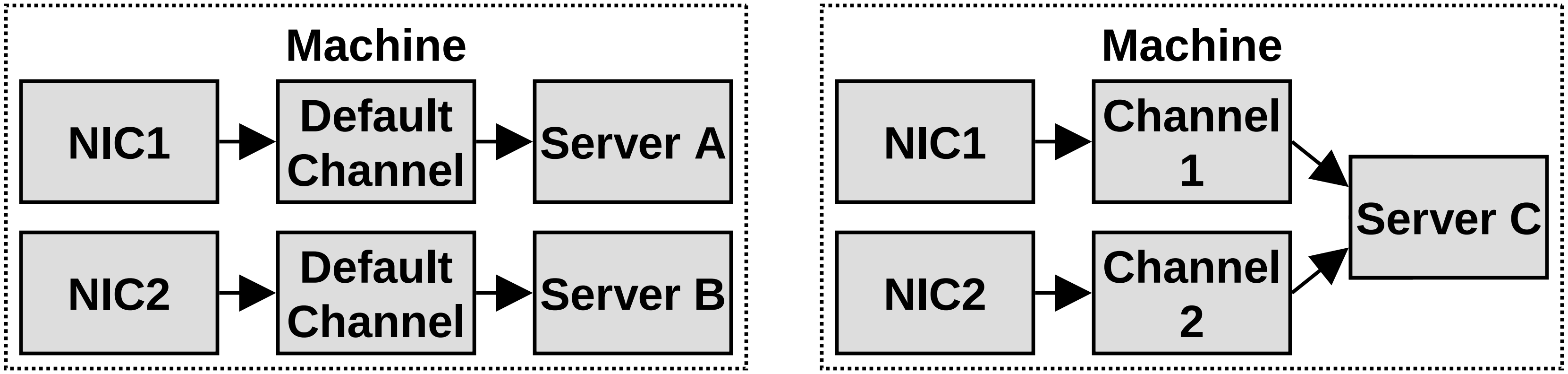
1. Remote Administration
2. Logs and Monitoring
3. SNMP Concepts
4. WLS SNMP Agent
5. WLS SNMP Management Tools
6. **Network Channels**
   - Addressing Features
   - Administration Port

Network Addressing Features…

- Adds flexibility to networking configuration:
  - multiple NICs for a single WLS server
  - specific NIC's or multiple port numbers on a NIC for specific WLS servers
  - multiple IP addresses can be used with each server
  - a single IP address can be used with multiple ports
  - configure the cluster multicast port number independently of the port numbers used by cluster members
  - multiple SSL configurations on one server

…Network Addressing Features

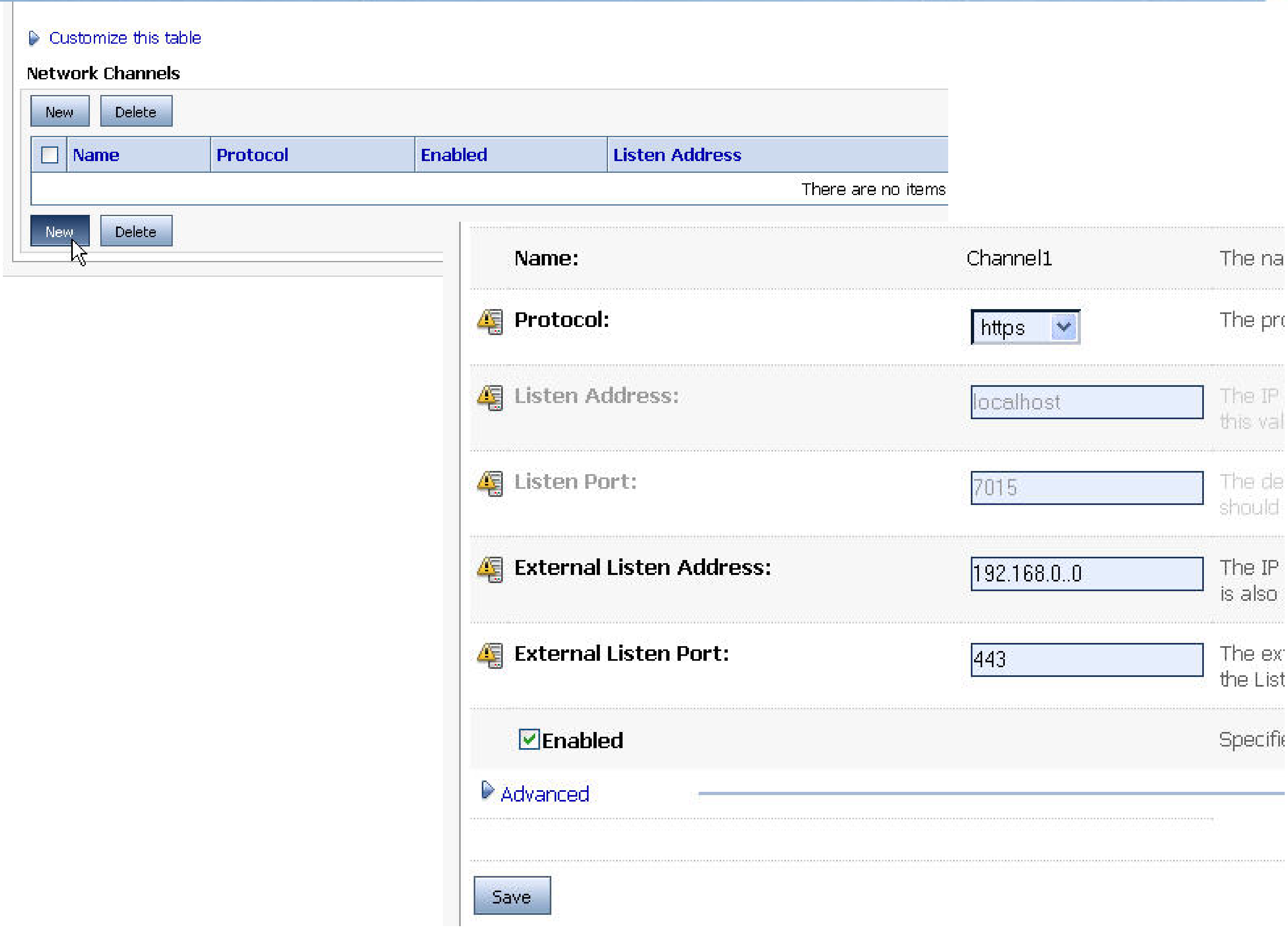
- Adds flexibility to networking configuration:
  - administration traffic only port
  - interoperability with previous WLS versions



Network Channels

- Network channels:
  - define a set of basic attributes of a network connection to WLS.
  - can assign multiple channels to a single server (segment network traffic).
  - can prioritize internal (non-URL) connections.
  - can separate incoming client traffic from internal server to server traffic in a domain.
  - “default” channel gets automatically generated when a server is created.

Configuring Network Channels



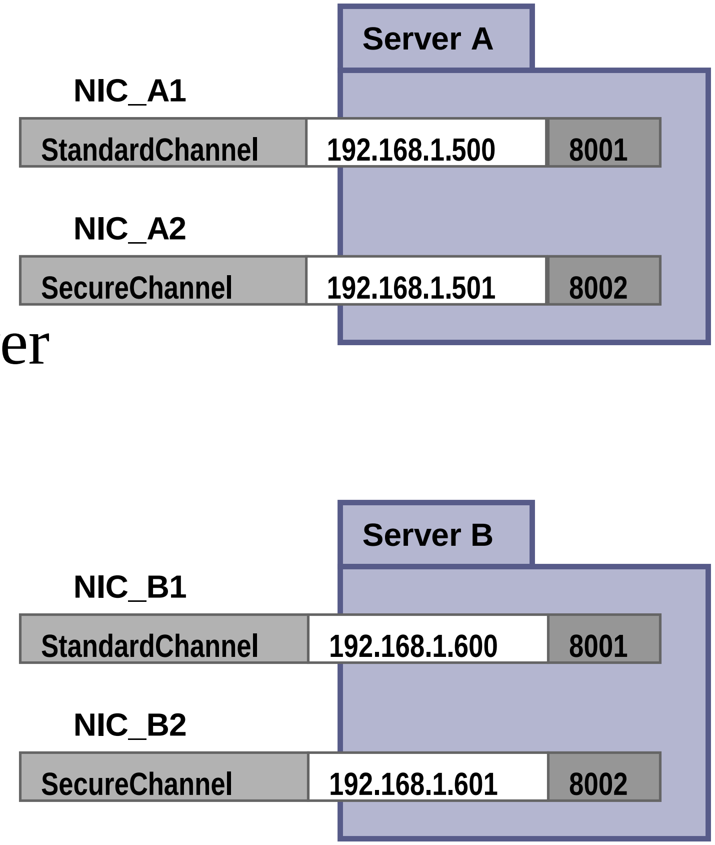
Network Channels

□	Name	Protocol	Enabled	Listen Address
				There are no items

Name:	Channel1	The na
Protocol:	https	The pr
Listen Address:	localhost	The IP this val
Listen Port:	7015	The de should
External Listen Address:	192.168.0..0	The IP is also
External Listen Port:	443	The ex the List
☑ Enabled		Specifi

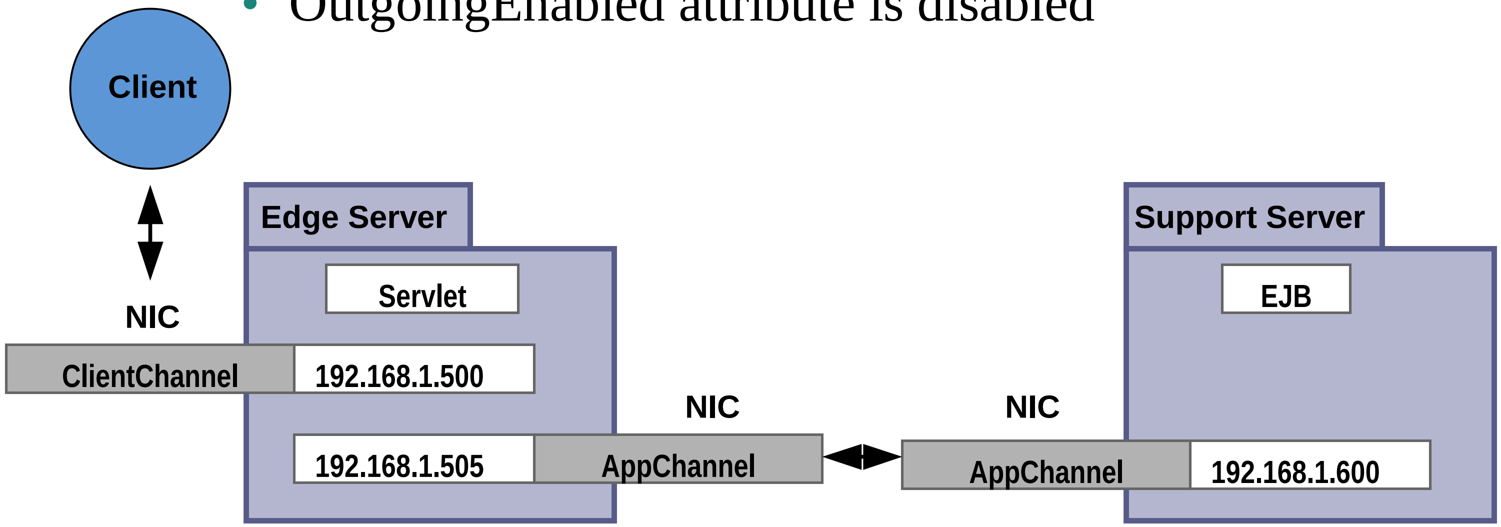
Using Channels Example 1

- Multiple NICs per server
  - each server has 2 NICs
  - Each NIC has one channel, hence there are 2 channels per server
  - Types of channels
    - StandardChannel
      - enables HTTP
      - disables other protocols
    - SecureChannel
      - enables HTTPS
      - disables other protocols



Using Channels Example 2

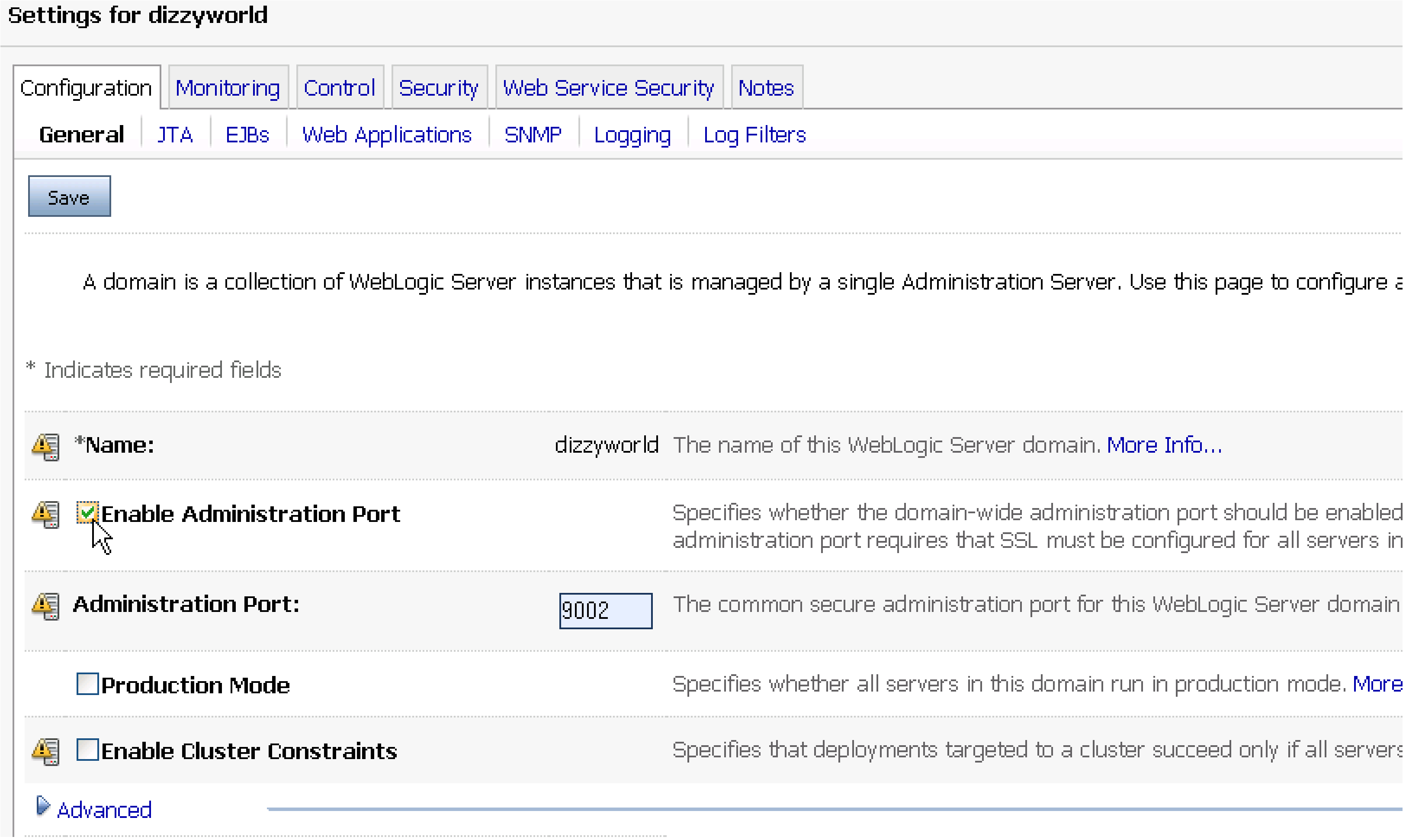
- Separate Internal and External traffic:
  - AppChannel is common between servers
    - used for internal communications
    - OutgoingEnabled attribute is enabled
  - ClientChannel is used for external access
    - clients can only connect to public IP address 192.168.1.500
    - OutgoingEnabled attribute is disabled



Administration Port…

- WLS allows configuration of a dedicated Administration Port:
  - generates an Administration channel
  - channel settings are as default channel except:
    - separate `SSLListenPort` value is defined
    - non-SSL `ListenPort` is disabled
  - only secure `t3s` and `https` admin traffic is allowed (no IIOP), only from:
    - console, weblogic.Admin and Managed Servers
  - all traffic requires two-way authentication
  - enables to start the server in Standby mode

…Administration Port



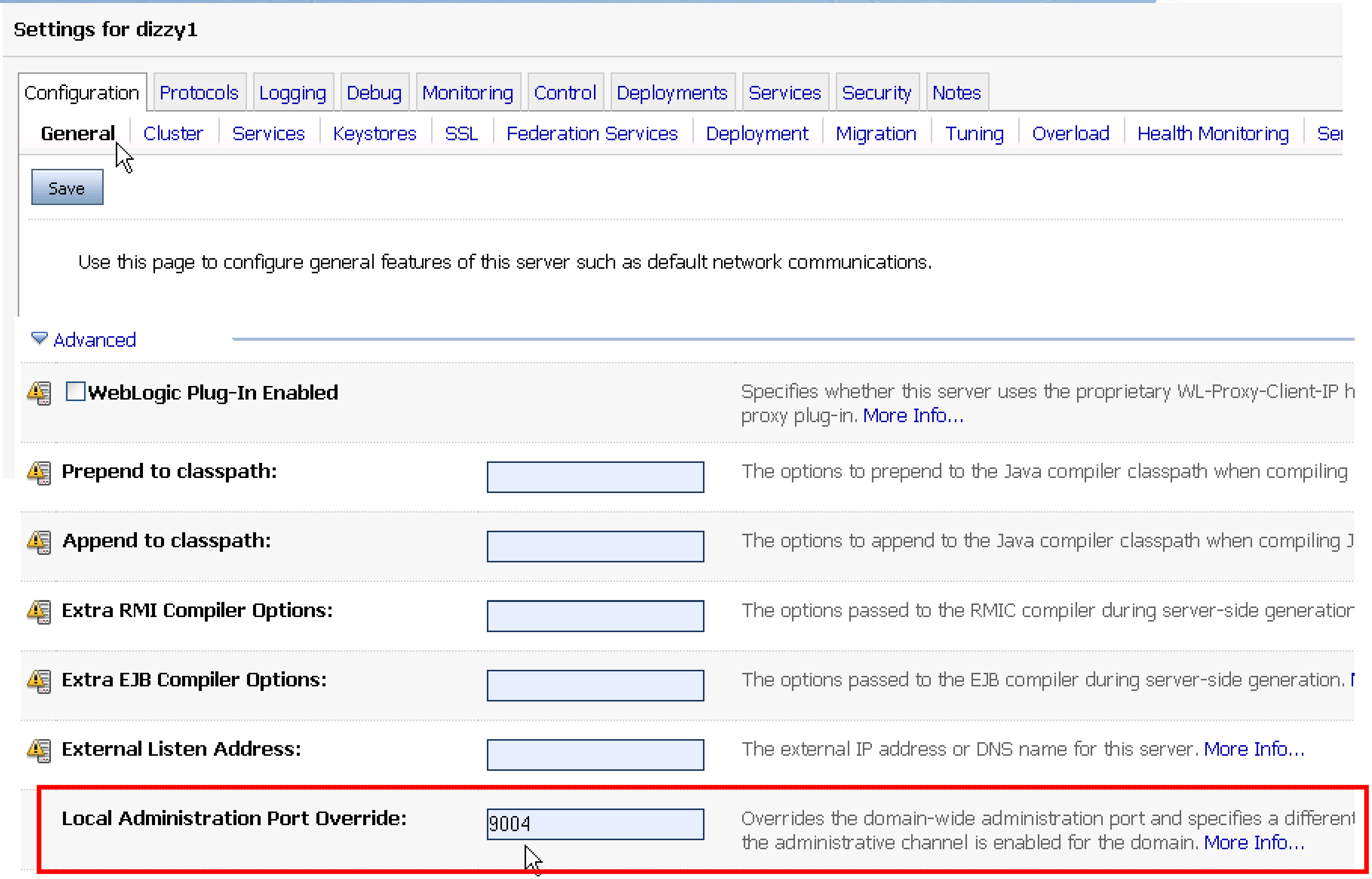
Settings for dizzyworld

A domain is a collection of WebLogic Server instances that is managed by a single Administration Server. Use this page to configure a

* Indicates required fields

*Name:	dizzyworld	The name of this WebLogic Server domain. More Info...
☑ Enable Administration Port		Specifies whether the domain-wide administration port should be enabled administration port requires that SSL must be configured for all servers in
Administration Port:	9002	The common secure administration port for this WebLogic Server domain
□ Production Mode		Specifies whether all servers in this domain run in production mode. More
□ Enable Cluster Constraints		Specifies that deployments targeted to a cluster succeed only if all servers

Override Administration Port



Settings for dizzy1

Use this page to configure general features of this server such as default network communications.

☐ **WebLogic Plug-In Enabled** — Specifies whether this server uses the proprietary WL-Proxy-Client-IP h proxy plug-in. More Info...

Prepend to classpath: — The options to prepend to the Java compiler classpath when compiling

Append to classpath: — The options to append to the Java compiler classpath when compiling J

Extra RMI Compiler Options: — The options passed to the RMIC compiler during server-side generatior

Extra EJB Compiler Options: — The options passed to the EJB compiler during server-side generation. I

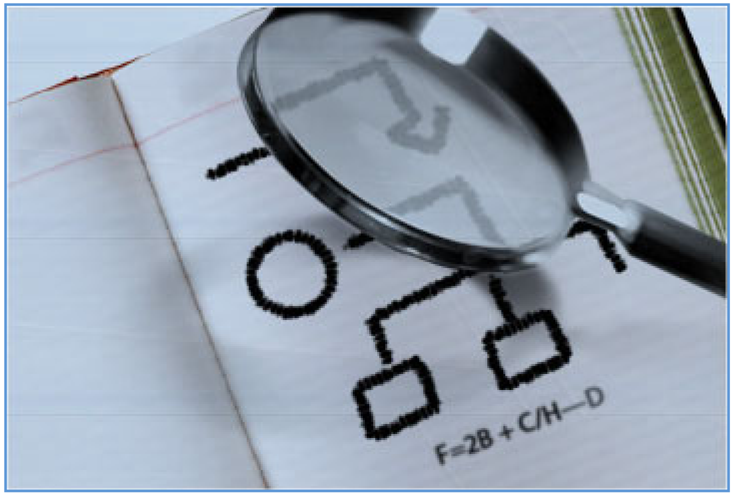
External Listen Address: — The external IP address or DNS name for this server. More Info...

Local Administration Port Override: 9004 — Overrides the domain-wide administration port and specifies a different the administrative channel is enabled for the domain. More Info...

Section Review

In this section we discussed:

- ✓ Network channels
- ✓ Administration Port



Exercise

Configuring Network Channels/Network Access Points

- In this lab you will configure Network Channels.
- Ask the instructor for any clarification.
- The instructor will determine the stop time.

Lab Exercise

Module Review

In this module we discussed:

- ✓ The benefits of Node Manager
- ✓ How to monitor domains and servers
- ✓ SNMP concepts
- ✓ The WLS SNMP Agent
- ✓ WLS-provided SNMP manager commands
- ✓ Configuring Network Channels