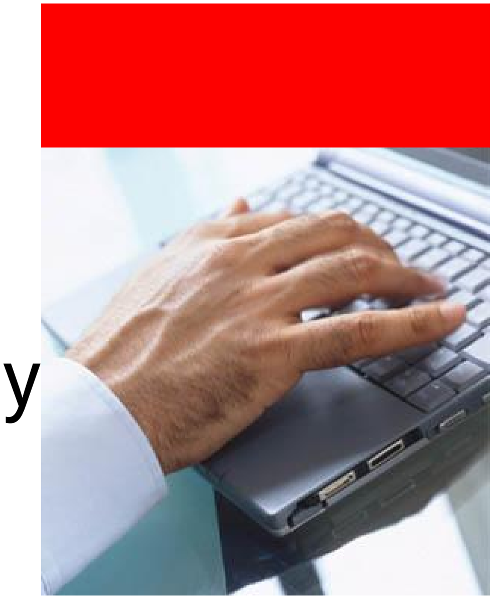
WebLogic Server: Installation and Configuration

Agenda

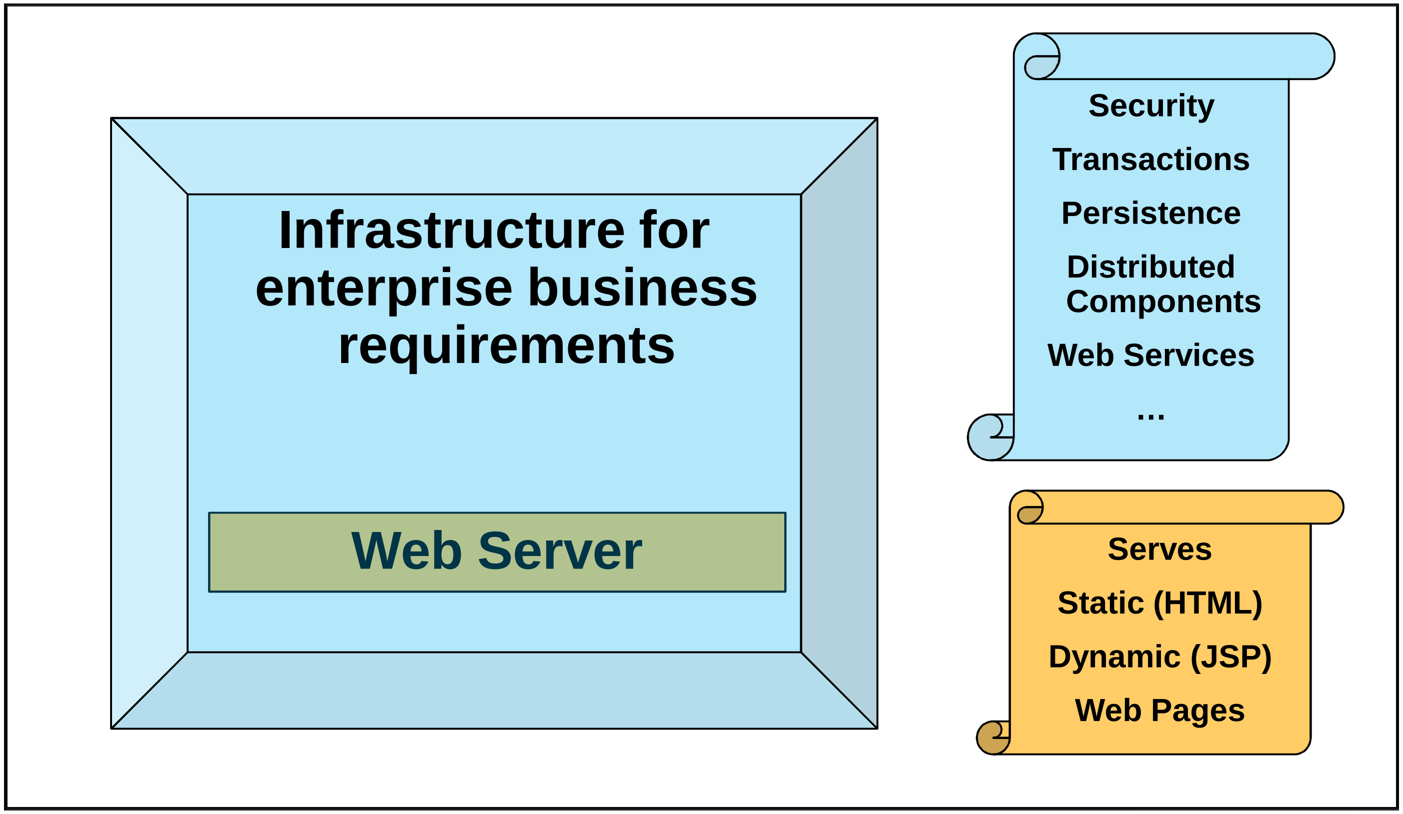
- Application server / Weblogic topology
- Download and Installation
- Configuration files. Demo
- Administration Tools: Configuration Wizard. Demo
- Administration Tools: Console. Demo
- Summary
- References
- Q&A

Agenda



- Application server / Weblogic topology
- Download and Installation
- Configuration files. Demo
- Administration Tools: Configuration Wizard. Demo
- Administration Tools: Console. Demo
- Summary
- References
- Q&A

Application server



Infrastructure for enterprise business requirements

Web Server

Security

Transactions

Persistence

Distributed Components

Web Services

…

Serves

Static (HTML)

Dynamic (JSP)

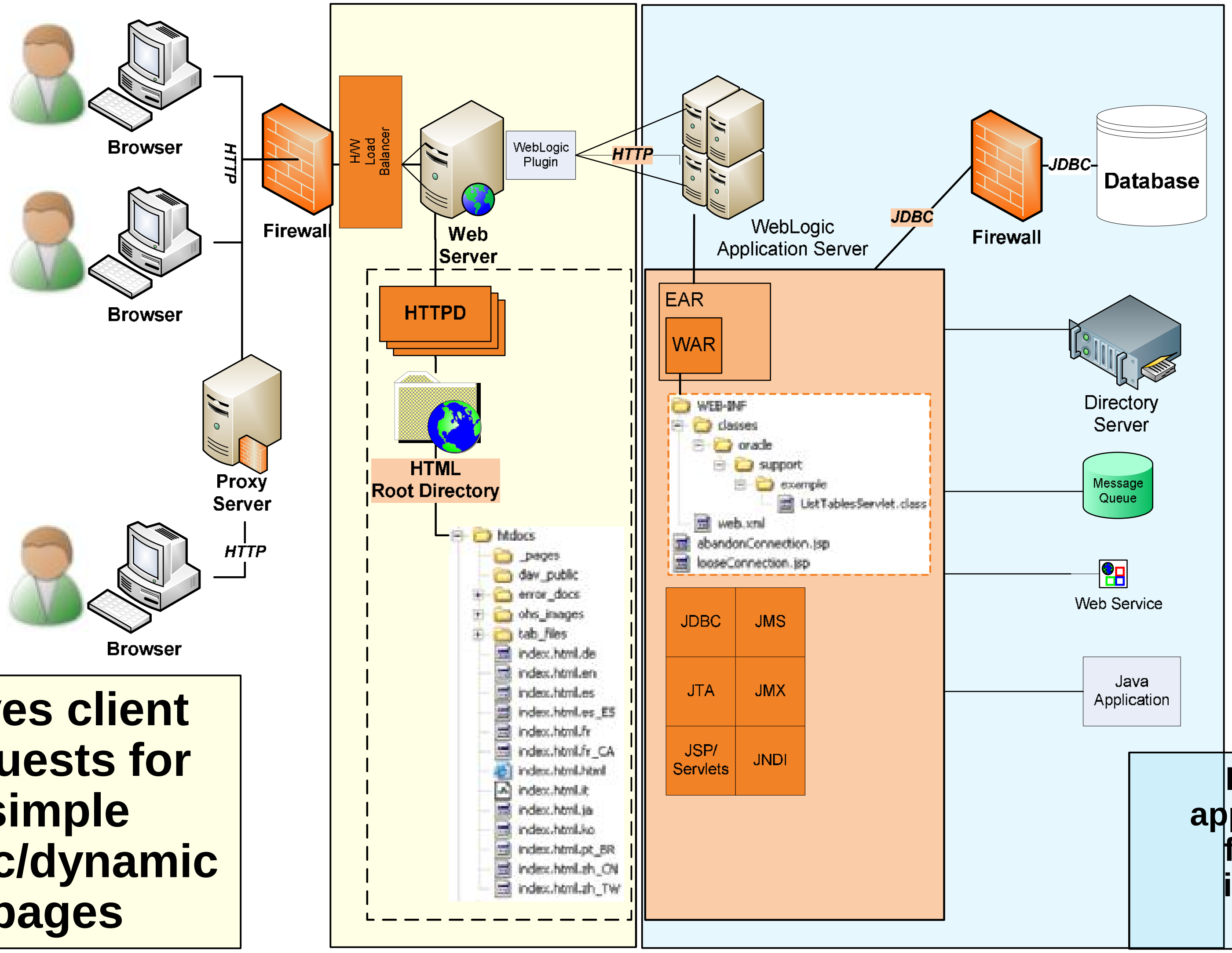
Web Pages

Oracle Weblogic Server is a J2EE application server It does NOT provide a full Web Server listener (e.g unlike Oracle Application Server which includes Oracle HTTP Server (based on Apache). Hence there is no HTTP Server shown in the topology diagram above.

There is, however, provide a lightweight listener. Plug-ins also exist to integrate Weblogic with 3rd Party listeners such as Apache. Arguably weblogic is closest in concept to an Oracle Application Server 10g Rel 2 J2EE Server only install.

Application server topology

Used in Fusion Applications

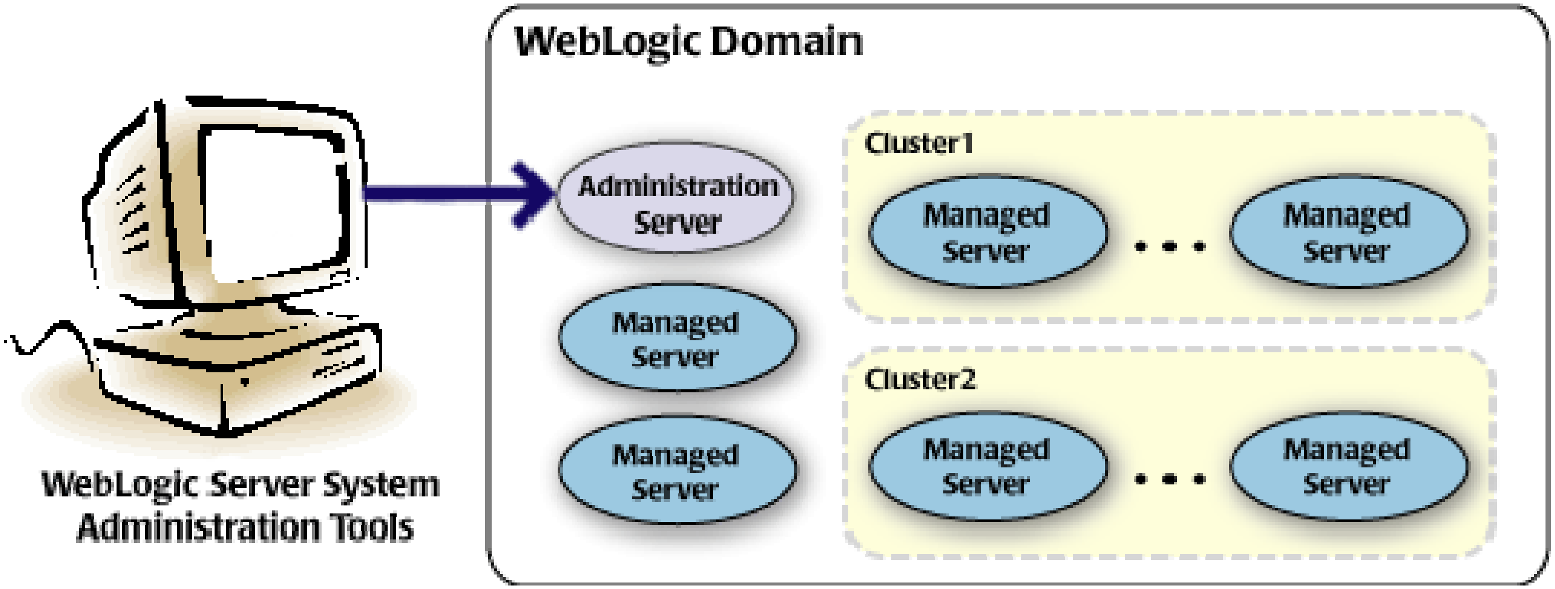


Serves client requests for simple static/dynamic pages

Forwards to application server for enterprise infrastructure support

Weblogic server topology

- Domain - group of servers and clusters under unified control
- Administration Server - central configuration controller for the domain
- Managed Server - an instance (similar in concept to an OracleAS 'OC4J' instance in which j2ee applications are deployed and configured.
- Cluster - group of managed servers to provide increased scalability and reliability
- Node Manager - not shown in the diagram, but this is 'OPMN' like per-machine process used to start, stop and auto-restart standalone managed servers and / or clusters

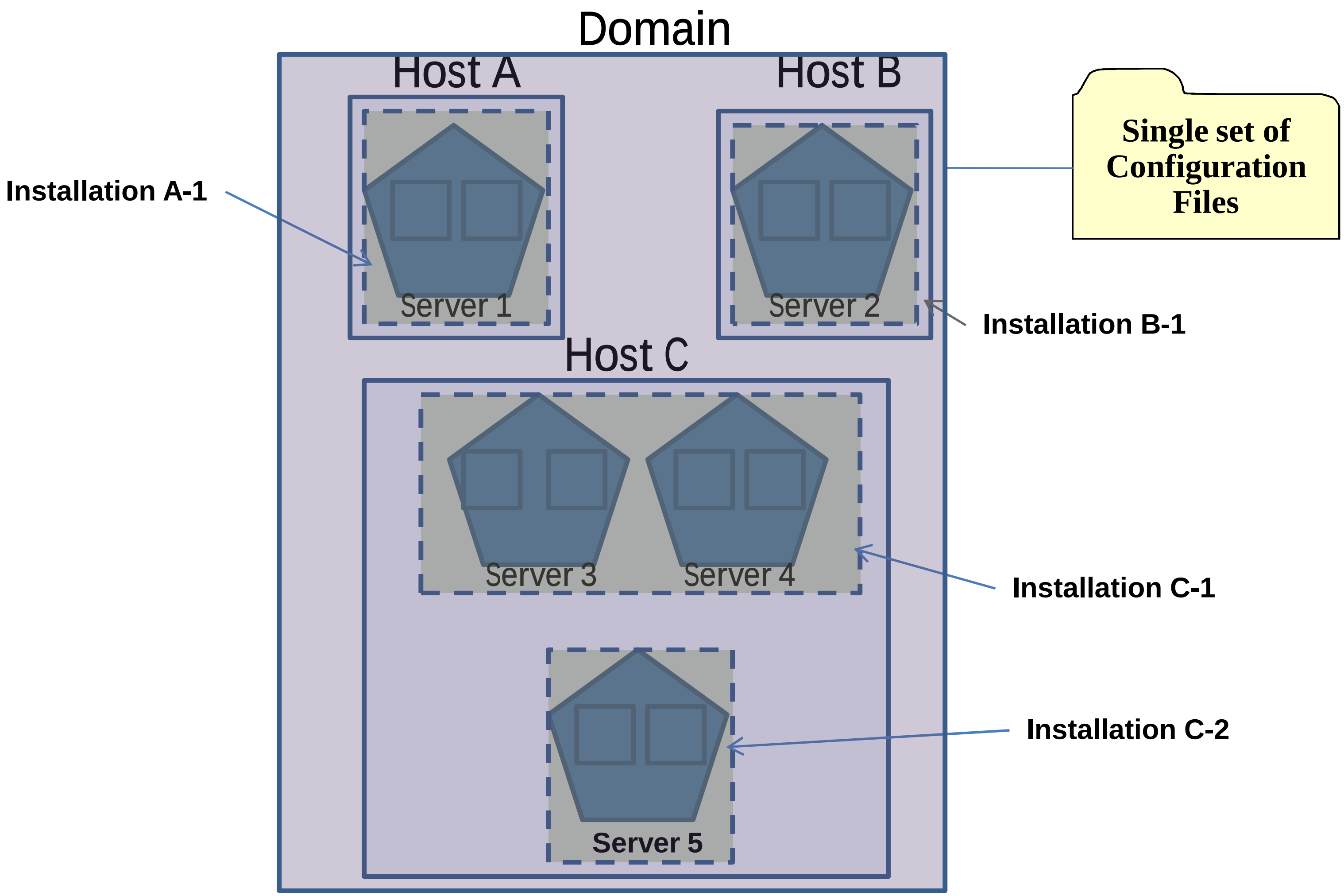


Domain

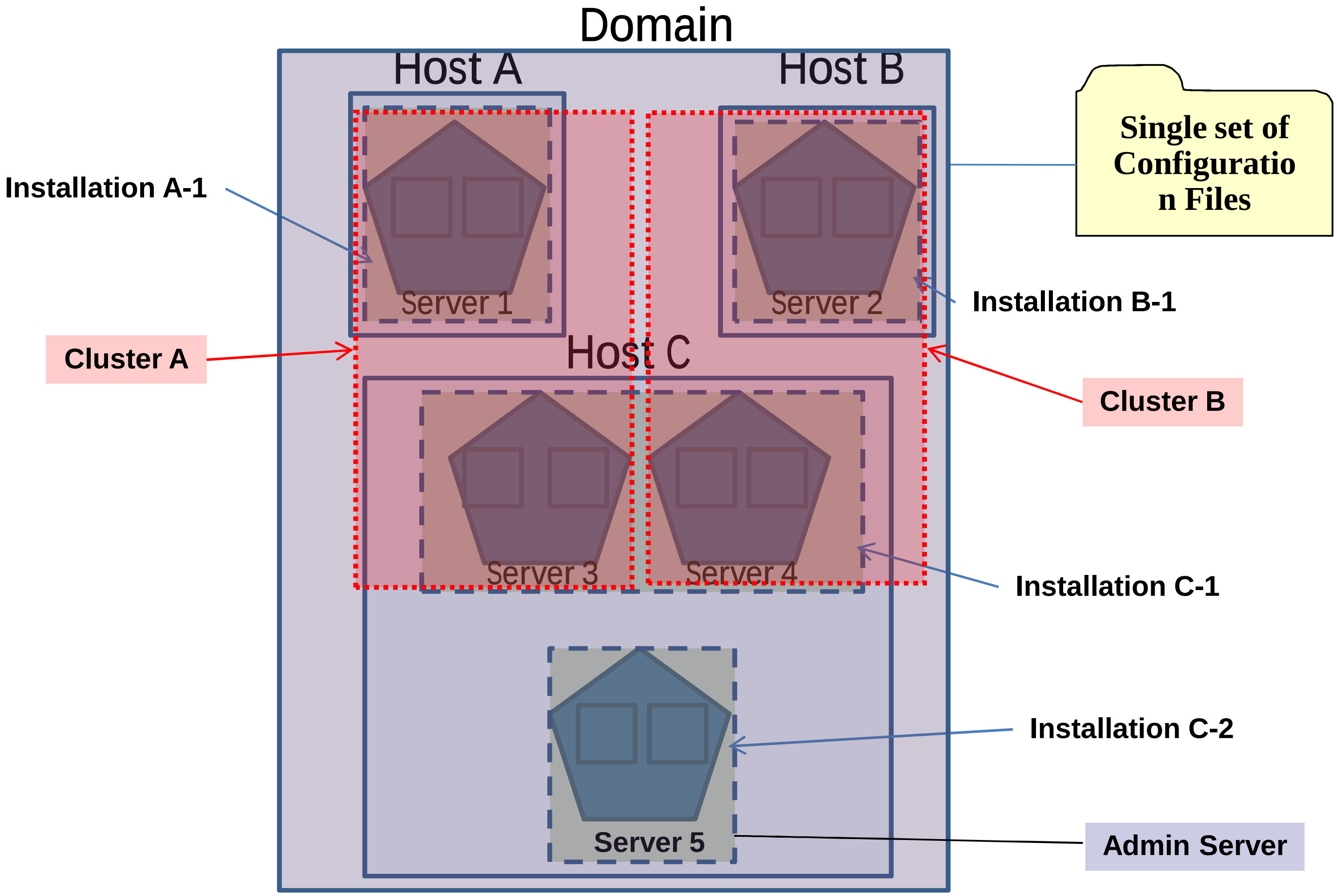
- A logically related group of WLS instances that you manage from a single set of configuration artifacts i.e. a single set of `config.xml` files
- What's in a domain?
  - Multiple Servers
    - Configuration pertaining to all the servers
- Servers in a domain could be
  - On the same or different machines
    - Using the same or different WLS installations

A domain is a collection of servers (whether standalone or part of a cluster). A domain can span a number of different physical machines but all within the domain is controlled via a ‘central’ administration server. The function of the domain is to make administration more straightforward and reliable. The concept of domain does not come into play during the runtime of applications.

Domain


Domain
Host A
Host B
Host C
Server 1
Server 2
Server 3
Server 4
Server 5
Installation A-1
Installation B-1
Installation C-1
Installation C-2
Single set of Configuration Files

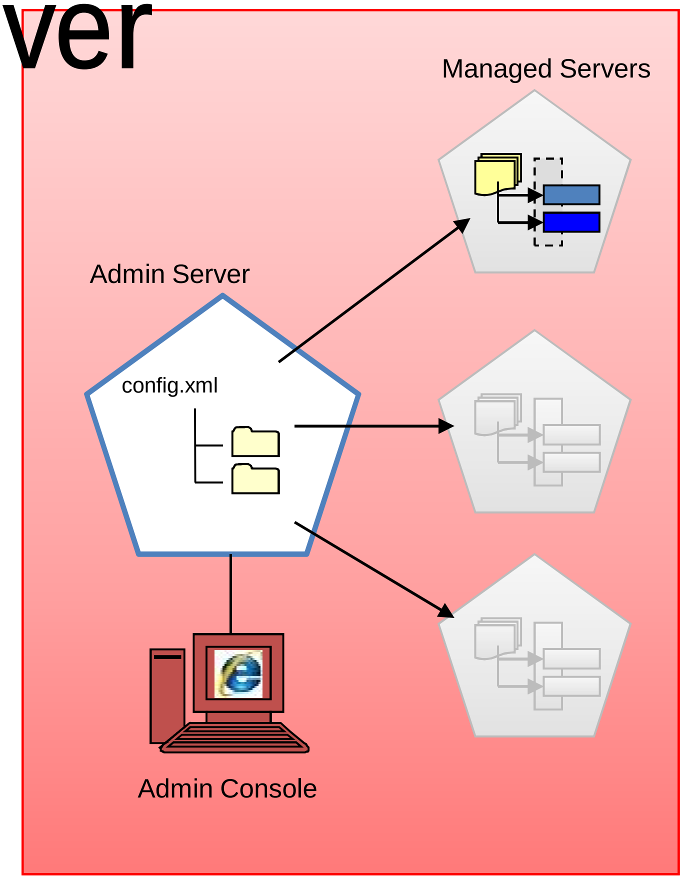
Domain with clusters



Administration Server

- What is it?
  - Central configuration controller for the entire domain
  - There is exactly one* Administration Server in domain
  - An Administration Server controls only one domain.
- Features
  - Hosts the Administration Console
    - which is a web application
  - Enables you to start and stop servers from a central location
  - Enables you to configure various services (Security, JDBC, JMS etc.) for the domain
  - Enables you to deploy applications within the domain

For production use, it is recommend to not host applications or resources on the Administration Server



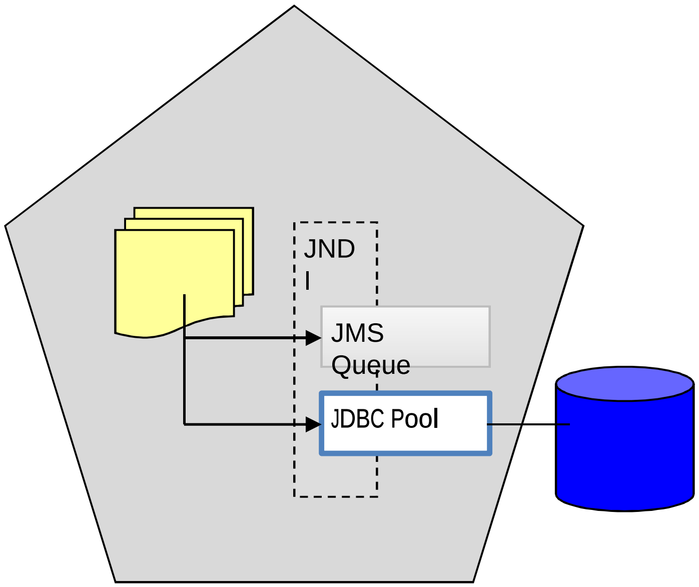
The Administration Server does not need to run at all times, but is required for making configuration and deployment changes to servers in the domain.

The Administration Server is a managed server instance which provides the configuration framework. Every domain must have its own administration server.

The Administration Server can be accessed through the WebLogic Server Scripting Tool and also from the Administration Console (the equivalent of OracleAS AS Control)

If the Administration Server is down the servers / clusters in its domain will work from a runtime perspective. In other words all servers will still be able to serve requests but configuration changes, deployments, etc will not be possible (similar to the way Oracle Application Server 10g behaves if the DCM-daemon is down / broken). This made possible by the Weblogic managed server keeping a local copy of its configuration so it can run in a “disconnected from the administration server” mode.

Managed server



- What is it?
  - A running instance that hosts applications and resources needed by those applications –
    - The real work horses in a WebLogic domain
  - Each Managed Server is independent of all other Managed Servers in the domain
  - You can have as many Managed Servers in a domain as you need
  - Individual Managed Servers are typically added for capacity and application isolation

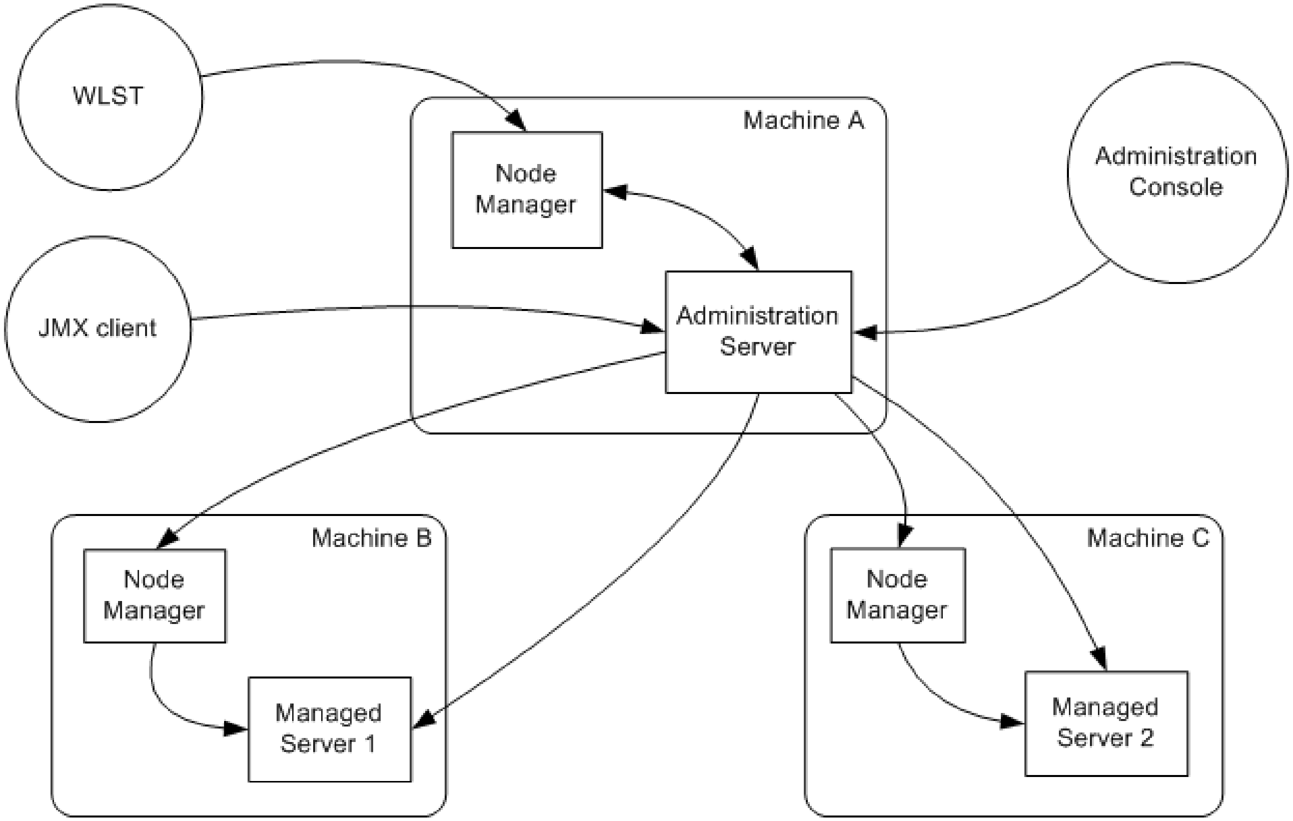
A significant difference between a weblogic managed server and an OC4J instance is that a weblogic server can only have one JVM. In an OracleAS 10g Rel 1 or 2 environment an OC4J instance can be configured to start up more than one JVM. To achieve the equivalent in Oracle Weblogic you would need to create a cluster of two or more weblogic managed servers e.g

OracleAS 10.1.3 J2EE only install with 1 OC4J instances (two islands i.e each one with 2 JVMs)
=> a single domain containing 1 * 2 = 2 servers in cluster

Node manager

- What is it?
  - The node manager starts up and runs as a separate java process from any managed server.
  - Can be used to start / stop and monitor servers and clusters within a single domain

- Node Manager is an optional tool.
- You can start/stop a Server without the Node Manager



The node manager is similar, although far less straightforward than, the OPMN component in Oracle Application Server. For example node manager can be configured to manage servers remotely

Interaction

Administration Server to Managed Server Interaction

- The Administration Server stores the master copy of the domain configuration, including the configuration for all managed servers in the domain
- Each Managed Server stores a local copy of its configuration.
- When a Managed Server starts, it connects to the Administration Server to synchronize the configuration
  - If Admin server is down, when managed server is started, it uses its local copy of configuration files
- When configuration is changed, the Administration Server sends changed configuration to Managed Servers

When a Managed Server starts up, it connects to the domain's Administration Server to synchronize its configuration document with the document that the Administration Server maintains.

Agenda

- Application server / Weblogic topology
- Download and Installation
- Configuration files. Demo
- Administration Tools: Configuration Wizard. Demo
- Administration Tools: Console. Demo
- Summary
- References
- Q&A

Download and Installation

- There are two places from which Oracle WebLogic Server can be downloaded.
  1. EDELIVERY: http://edelivery.oracle.com
  2. OTN: http://www.oracle.com/technology/software/products/ias/bea_main.html

Download and Installation

- How to install?
  - The downloaded zip contains a very large .exe or .bin file. The 'simplest' way to install is to run the installer in graphical mode.

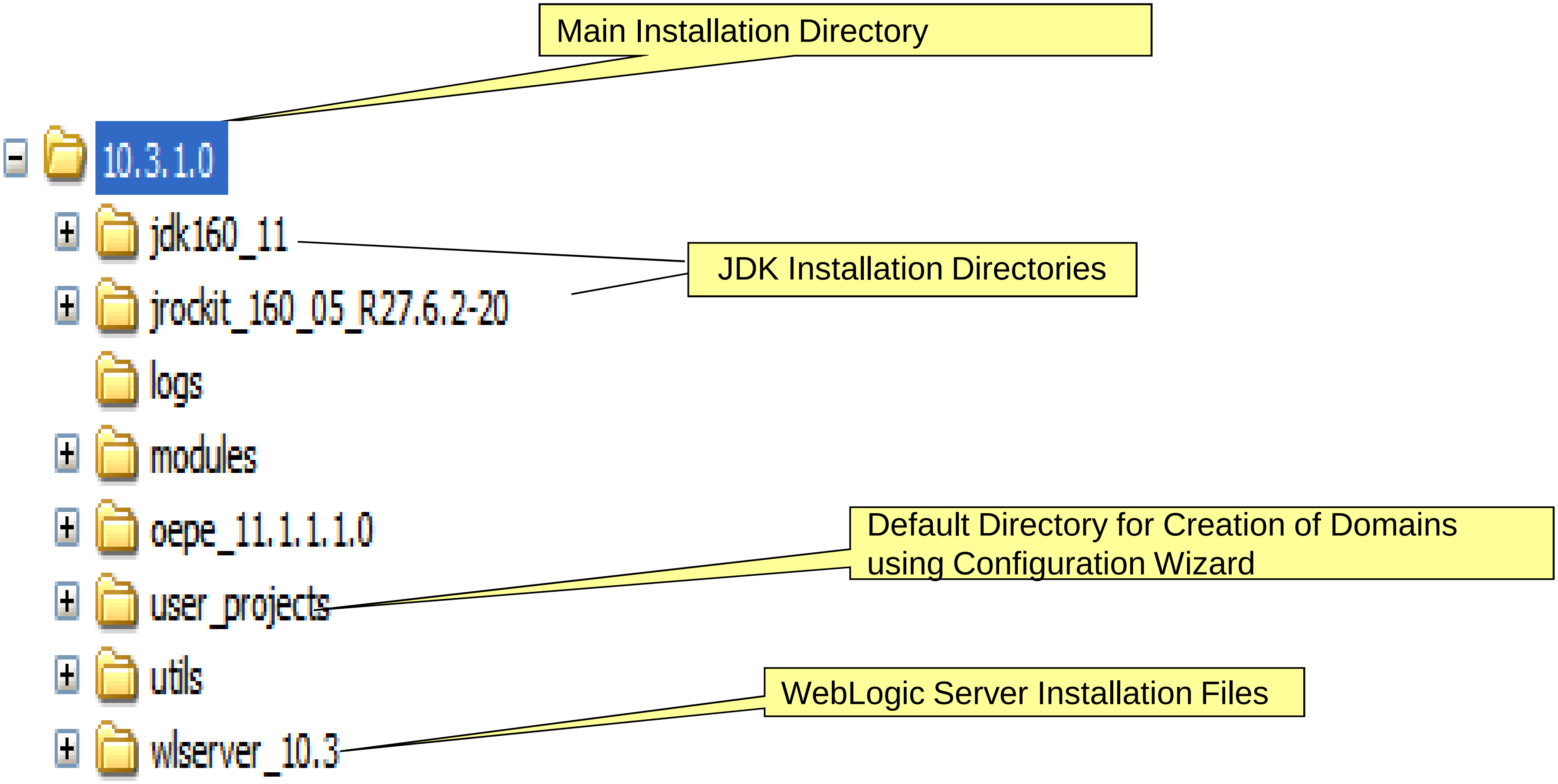
  For Unix:

  Run these commands to launch the graphical installer.

  ```
  chmod a+x <filename.bin>
  ./<filename.bin> -log=/home/logs/WLS_install.log
  ```

- It is also possible to launch the install in two other modes:
  - Console Mode - a non-graphical interactive install during which the installer will prompt for information / answers to questions via the command line
  - Silent Mode - a non-interactive install, installation options are specified in a .xml properties file.

Directory structure – installation



- 10.3.1.0
  - jdk160_11
  - jrockit_160_05_R27.6.2-20
  - logs
  - modules
  - oepe_11.1.1.1.0
  - user_projects
  - utils
  - wlserver_10.3

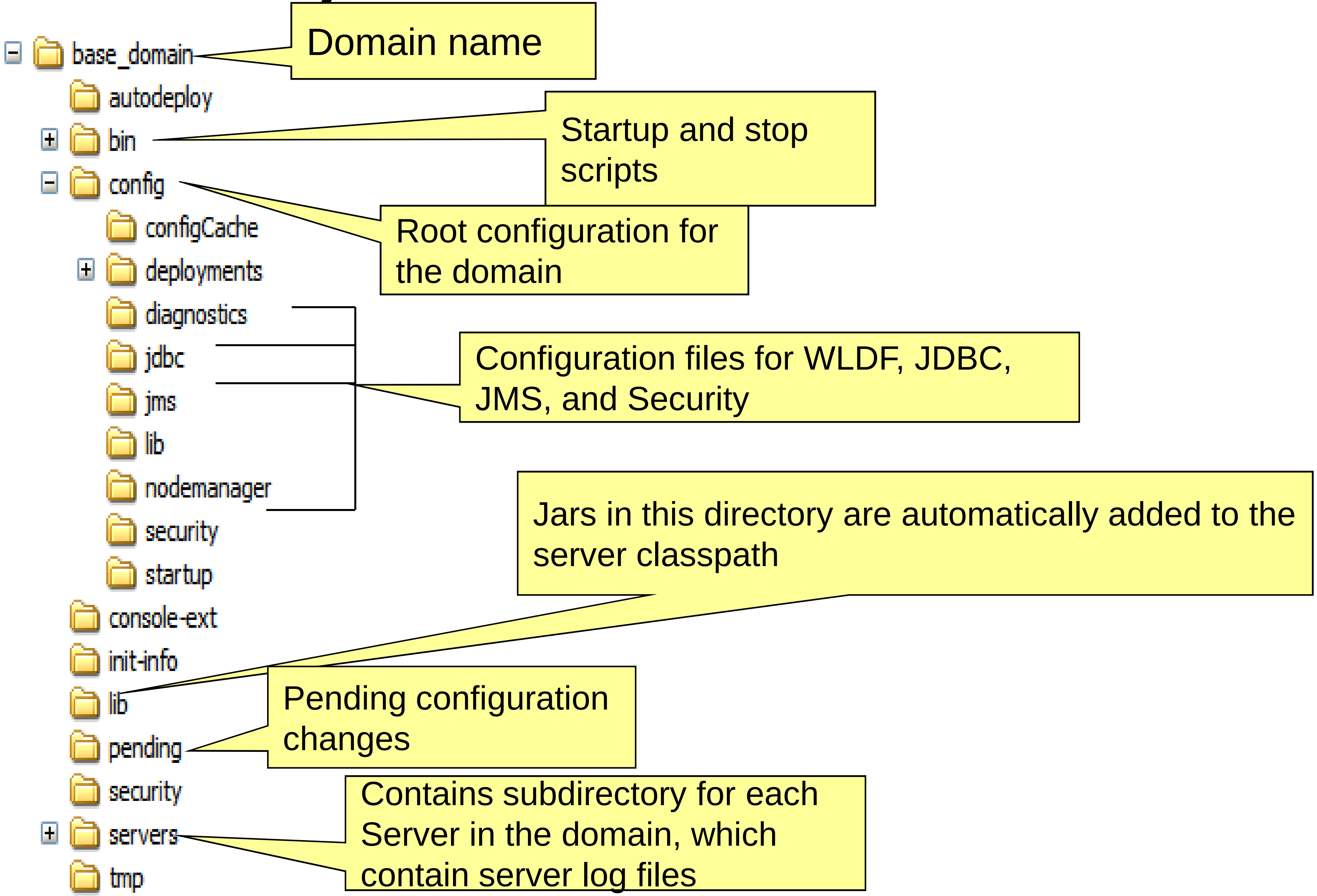
Main Installation Directory

JDK Installation Directories

Default Directory for Creation of Domains using Configuration Wizard

WebLogic Server Installation Files

Directory structure – domain



```
⊟ base_domain
    autodeploy
  ⊞ bin
  ⊟ config
      configCache
    ⊞ deployments
      diagnostics
      jdbc
      jms
      lib
      nodemanager
      security
      startup
    console-ext
    init-info
    lib
    pending
    security
  ⊞ servers
    tmp
```

Domain name

Startup and stop scripts

Root configuration for the domain

Configuration files for WLDF, JDBC, JMS, and Security

Jars in this directory are automatically added to the server classpath

Pending configuration changes

Contains subdirectory for each Server in the domain, which contain server log files *domain-name*: **The name of this directory is the name of the domain.**

Autodeploy: **in development mode, apps and files are automatically deployed**

Bin: **contains scripts that are used in the process of starting and stopping the Administration Server and the Managed Servers in the domain. can optionally contain other scripts of domain-wide interest,**

config: **contains the current configuration and deployment state of the domain. The central domain configuration file, config.xml, resides in this directory.**

config/configCache: **Contains data that is used to optimize performance when validating changes in the domain's configuration documents.**

config/diagnostics: **contains system modules for the WebLogic Diagnostic Framework.**

config/jdbc: **contains system modules for global JDBC modules that can be configured directly from JMX**

config/jms: **contains system modules for JMS**

config/lib: **not used in the current release of WebLogic Server.**

config/nodemanager: **holds configuration information for connection to the Node Manager.**

config/security: **contains system modules for the security framework. It contains one security provider configuration extension for each kind of security provider in the domain's current realm.**

config/startup: **contains system modules that contain startup plans. Startup plans are used to generate shell scripts that can be used as part of server startup.**

configArchive: **contains a set of JAR files that save the domain's configuration state. Just before pending changes to the configuration are activated, the domain's existing configuration state, consisting of the config.xml file and the other related configuration files, is saved in a versioned JAR file with a name like config.jar#1, config.jar#2, etc.**

console-ext: **contains extensions to the Administration Console, which enable you to add content to the WebLogic Server Administration Console, replace content, and change the logos, styles and colors without modifying the files that are installed with WebLogic Server.**

init-info: **contains files used for WebLogic domain provisioning.**

lib: **Any JAR files you put in this directory are added to the system classpath of each server instance in the domain when the server's Java virtual machine starts.**

pending: **contains domain configuration files representing configuration changes that have been requested, but not yet activated.**

security: **holds security-related files that are the same for every WebLogic Server instance in the domain:**

SerializedSystemIni.dat

This directory also holds security-related files that are only needed by the domain's Administration Server:

DefaultAuthorizerInit.ldift

DefaultAuthenticatorInit.ldift

DefaultRoleMapperInit.ldift

servers: **contains one subdirectory for each WebLogic Server instance in the domain. The subdirectories contain data that is specific to each server instance.**

servers/*server-name:* **the server directory for the WebLogic Server instance with the same name as the directory.**

servers/*server-name*/bin: **holds executable or shell files that can be or must be different for each server.**

servers/*server-name*/cache: **holds directories and files that contain cached data.**

servers/*server-name*/cache/EJBCompilerCache: **cache for compiled EJBs.**

servers/*server-name*/data: **holds files that maintain persistent per-server state used to run the WebLogic Server instance, other than security state, as opposed to temporary, cached or historical information.**

servers/*server-name*/data/ldap: **holds the embedded LDAP database. The runtime security state for the WebLogic Server instance is persisted in this directory.**

servers/*server-name*/data/store: **holds WebLogic persistent stores.**

servers/*server-name*/logs: **holds logs and diagnostic information.**

servers/*server-name*/logs/diagnostic_images: **holds information created by the Server Image Capture component of WLDF**

servers/*server-name*/logs/jmsServers: **contains one subdirectory for each JMS server in the WebLogic Server instance. Each such subdirectory contains the logs for that JMS server.**

servers/*server-name*/logs/connector: **the default base directory for connector module (JCA ResourceAdapter) logs.**

servers/*server-name*/security: h**olds security-related files that can be or must be different for each WebLogic Server instance. The file boot.properties is an example of a file that resides here because it can differ from one server to the next. This directory also maintains files related to SSL keys.**

servers/*server-name*/tmp

This directory holds temporary directories and files that are created while a server instance is running. For example, a JMS paging directory is automatically created here unless another location is specified. Files in this directory must be left alone while the server is running, but may be freely deleted when the server instance is shut down.

tmp: **stores temporary files used in the change management process.**

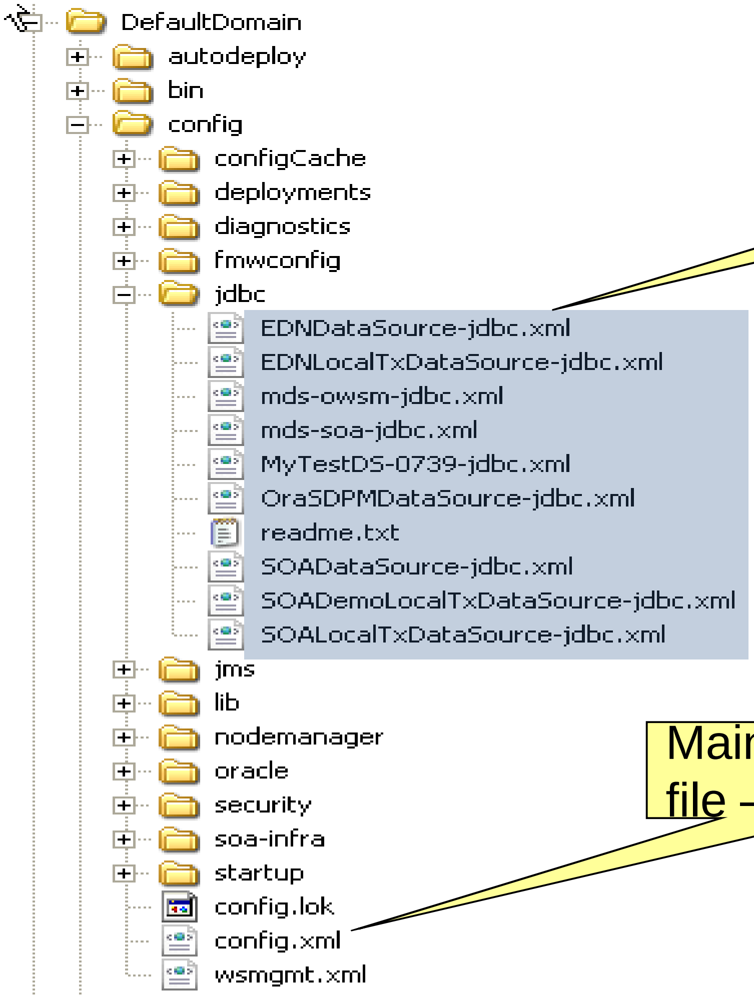
Agenda

- Application server / Weblogic topology
- Download and Installation
- Configuration files. Demo
- Administration Tools: Configuration Wizard. Demo
- Administration Tools: Console. Demo
- Summary
- References
- Q&A

Configuration files

- One central configuration file for a domain
  - `config.xml`
  - Located in `<Domain_Directory>/config` directory
- Includes the configuration of each server instance, cluster, resource, and service in the domain.
- References additional XML files that are stored in subdirectories of the `<Domain_Directory>/config` directory : `jms`, `jdbc, wldf`, and `security`
- All files are based on XML Schemas

Configuration files



```
DefaultDomain
    autodeploy
    bin
    config
        configCache
        deployments
        diagnostics
        fmwconfig
        jdbc
            EDNDataSource-jdbc.xml
            EDNLocalTxDataSource-jdbc.xml
            mds-owsm-jdbc.xml
            mds-soa-jdbc.xml
            MyTestDS-0739-jdbc.xml
            OraSDPMDataSource-jdbc.xml
            readme.txt
            SOADataSource-jdbc.xml
            SOADemoLocalTxDataSource-jdbc.xml
            SOALocalTxDataSource-jdbc.xml
        jms
        lib
        nodemanager
        oracle
        security
        soa-infra
        startup
        config.lok
        config.xml
        wsmgmt.xml
```

JDBC specific Configuration files

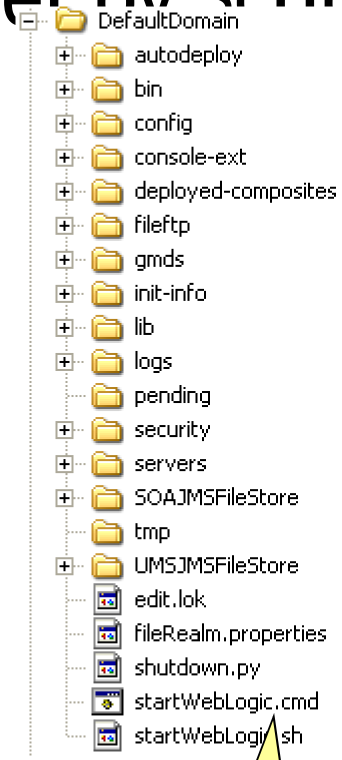
Main Configuration file – config.xml

Two-phase configuration changes

- Changes are made in two phases as in a database
  - Prepare
  - Commit
- Ensures reliability, consistency
- Edit Lock mechanism
  - Only one user can make a change at any point of time
- Process
  - Get an edit lock
  - Make changes
    - changes are stored in the `“<Domain_Directory>/pending”` directory
  - Activate your changes
    - Implicit validation takes place
    - Changes are distributed to servers in the domain
    - Two phases: Prepare and Commit
    - Prepared on all servers; any failures will cause total rollback

Starting/Stopping Server by Scripts

- Scripts can be used to start/stop servers
  - Provided by default in the `<Domain_Directory>`
- 2 versions of scripts:
  - Unix **à** `*.sh` files
  - Windows **à** `*.cmd` files
- Server startup scripts:
  - Admin Server:
    - `<Domain_Directory>/startWebLogic.sh OR`
    - `<Domain_Directory>/bin/startWebLogic.sh`
  - Managed Server:
    - `<Domain_Directory>/bin/startManagedWebLogic.sh`
- Server stop scripts:
  - Admin Server:
    - `<Domain_Directory>/bin/stopWebLogic.sh`
  - Managed Server:
    - `<Domain_Directory>/bin/stopManagedWebLogic.sh`



```
DefaultDomain
    autodeploy
    bin
    config
    console-ext
    deployed-composites
    fileftp
    gmds
    init-info
    lib
    logs
    pending
    security
    servers
    SOAJMSFileStore
    tmp
    UMSJMSFileStore
    edit.lok
    fileRealm.properties
    shutdown.py
    startWebLogic.cmd
    startWebLogic.sh
```

Startup script file

Starting/Stopping server by scripts

Starting Managed Server

- Use the script
`<Domain_Directory>/bin/startManagedWebLogic.sh`
- Takes two parameters
  - Managed Server name
  - Admin Server URL **(**default `http://localhost:7001`**)**

Example: `startManagedWebLogic.sh MS1 http://localhost:7001`

- Credentials providing mechanism similar to Admin server startup

The script to start any Managed Server is:

$DOMH/bin/startManagedWebLogic.sh

This script takes 2 parameters:

Server Name - which bizzarly default to the AdminServer

Admin URL - which defaults to t3://localhost:7001

So running this script with no parameters is the same as running “startWeblogic.sh” and will try and start the AdminServer.

Starting any Managed Server demands the username/password as a prompt. The easiest way to stop this prompt is to store these values in the file $DOMH/servers/AdminServer/security/ boot.properties. The file contents would start like:

username=weblogic

password=weblogic

But after a first startup, both entries are then encrypted.

As an alternative, environment variables WLS_USER and WLS_PASSWORD can be set within the script startManagedWebLogic.sh.

It is not possible to pass these parameters on the command line.

Starting/Stopping server by scripts

Stopping Managed Server

- Use the script
  - `<Domain_Directory>/bin/stopManagedWebLogic.sh`
  - Takes following parameters
    - `Managed Server name`
    - `Admin Server URL` (default `http://localhost:7001`)
    - `username`
    - `password`

Example:

```
stopManagedWebLogic.sh MS1 http://localhost:7001 weblogic weblogic1
```

- Credentials providing mechanism similar to Admin server stop

The script $DOMH/bin/stopWebLogic.sh takes 3 parameters but all are optional:

UserName

Password

Admin URL - which defaults to t3://localhost:7001

Stopping the Admin Server will prompt for user/password if you have not supplied them or the file boot.properties does not exist.

Starting/Stopping server by scripts

Killing the JVM

- You can stop the servers by killing the JVM
  - Typically used only in Development environments
  - Windows
    - Do `Ctrl-C` on the command window used to start the server OR
    - Kill the process using Task Manager
  - Unix
    - Use `ps` command to list all processes
      - Get the Process id (PID) of the running WLS instance
    - Use “`kill -9 <PID>`” to kill the process

The script $DOMH/bin/stopWebLogic.sh takes 3 parameters but all are optional:

UserName

Password

Admin URL - which defaults to t3://localhost:7001

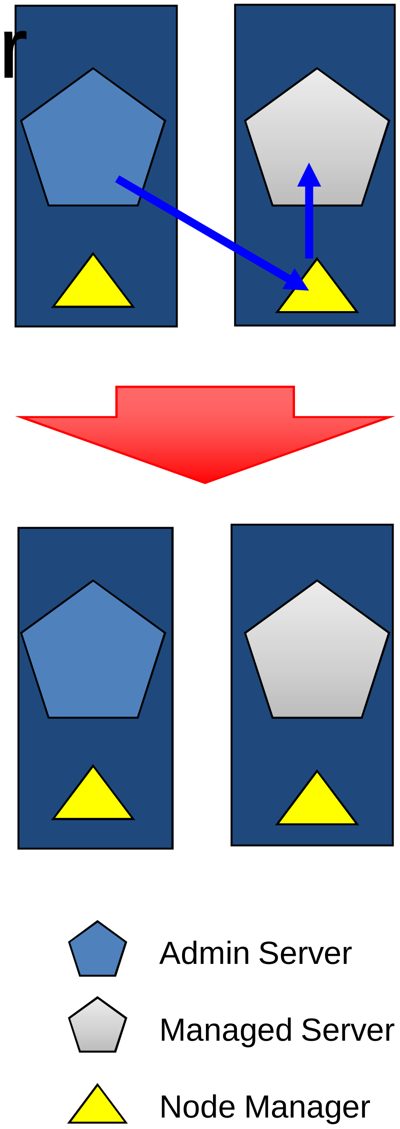
Stopping the Admin Server will prompt for user/password if you have not supplied them or the file boot.properties does not exist.

Starting/Stopping servers remotely

- Multiple Servers part of a domain could be on different physical machines
- Using startup script files, you have to go to access the remote machine and run the scripts
- Can we start/stop a server remotely? Through the web?

Node Manager

- Utility/process running on a physical server that enables you to start, stop, suspend, and restart WebLogic Server instances remotely
- Must run on each physical server that hosts WebLogic Server instances that you want to control with Node Manager
- Not associated with a domain. Can start any server instance that resides on the same physical server.
- Optional, but required to start/stop servers using the Administration Console



⬟ Admin Server

⬠ Managed Server

△ Node Manager

Server instances in a WebLogic Server production environment are often distributed across multiple domains, machines, and geographic locations. Node Manager is a WebLogic Server utility that enables you to start, shut down, and restart Administration Server and Managed Server instances from a remote location. Although Node Manager is optional, it is recommended if your WebLogic Server environment hosts applications with high availability requirements.

A Node Manager process is not associated with a specific WebLogic domain but with a machine. You can use the same Node Manager process to control server instances in any WebLogic Server domain, as long as the server instances reside on the same machine as the Node Manager process. Node Manager must run on each computer that hosts WebLogic Server instances -- whether Administration Server or Managed Server -- that you want to control with Node Manager.

Demo

Start WLS with scripts

Stop WLS with scripts

Kill JVM

http://efops.us.oracle.com:7779/pls/report/f?p=102:31:2066375453184218

1. Start AdminServer

```
D:\Middleware\WLS\user_projects\domains\T1> startWebLogic.cmd
weblogic/weblogic1
```

http://localhost:7011/console

2. Start managed server MS1

```
D:\Middleware\WLS\user_projects\domains\T1\bin> startManagedWebLogic.cmd T1_MS1
weblogic/weblogic1
```

Refresh in console

3. Stop T1_MS1 from console

4. Stop AdminServer using stopWebLogic.cmd

```
D:\Middleware\WLS\user_projects\domains\T1\bin> stopWebLogic.cmd
weblogic/weblogic1
```

Agenda

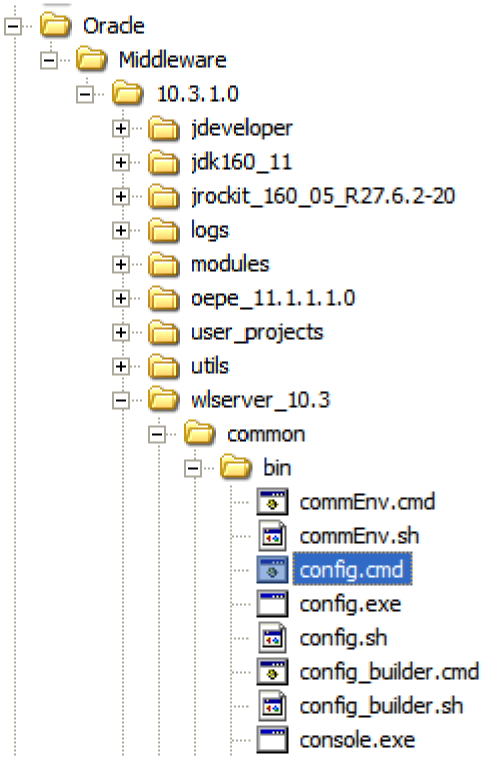
- Application server / Weblogic topology
- Download and Installation
- Configuration files. Demo
- Administration Tools: Configuration Wizard. Demo
- Administration Tools: Console. Demo
- Summary
- References
- Q&A

Configura

If you have JDeveloper installed along with WLS, you can also find the config wizard at

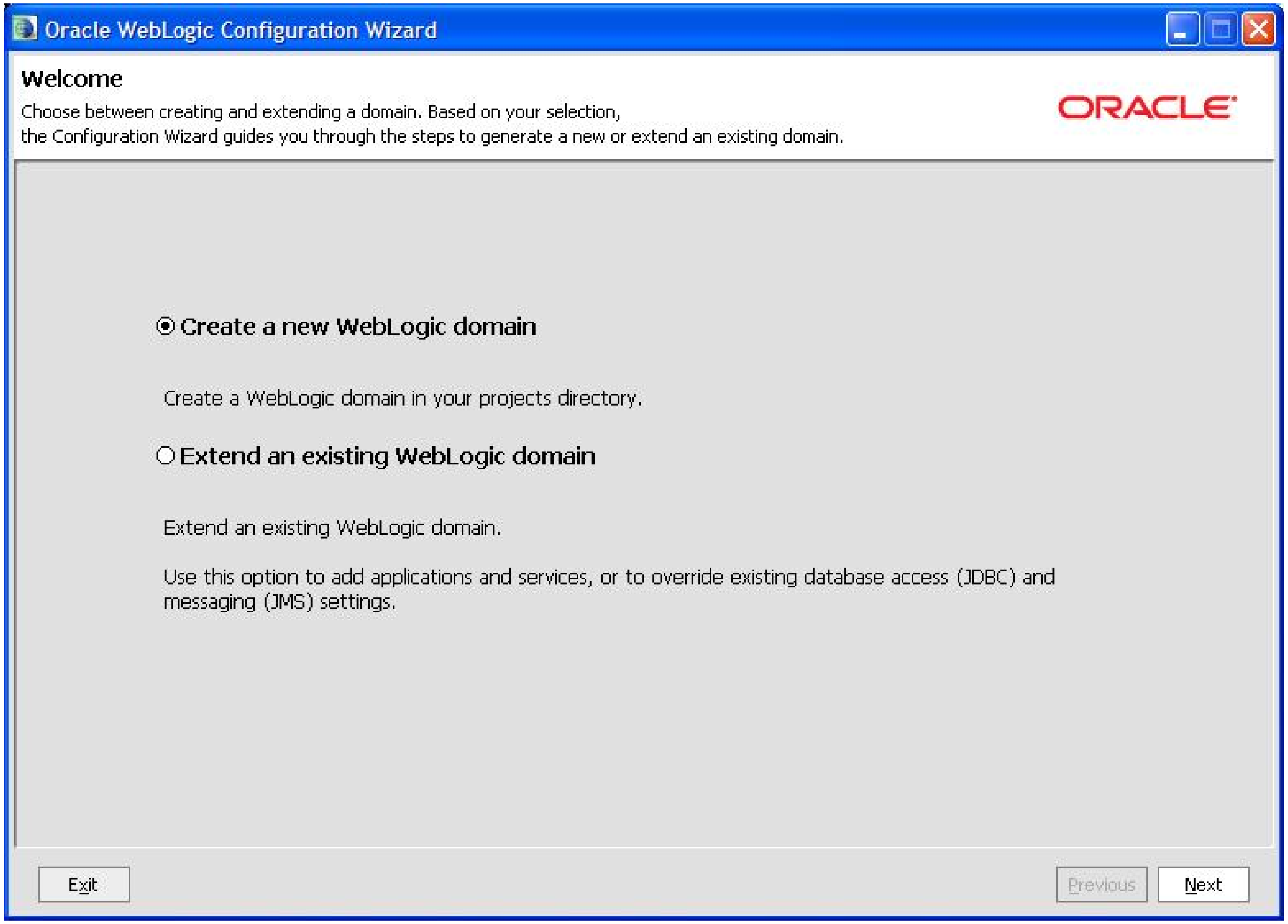
`<WLS_Install_Directory>/wlserver_10.3/jdeveloper/common/bin/config.sh`, which lets you create domains complaint with JDeveloper and ADF

- Wizard that walks you through domain creation off line
  - Available at
    - `<WLS_Install_Directory>/wlserver_10.3/common/bin/config.sh`
- Requires a domain *template*
  - A template defines the available settings you can make for a new domain
- WLS ships with a default template and templates for samples domains
- Customers and layered products can create their own templates



```
Oracle
  Middleware
    10.3.1.0
      jdeveloper
      jdk160_11
      jrockit_160_05_R27.6.2-20
      logs
      modules
      oepe_11.1.1.1.0
      user_projects
      utils
      wlserver_10.3
        common
          bin
            commEnv.cmd
            commEnv.sh
            config.cmd
            config.exe
            config.sh
            config_builder.cmd
            config_builder.sh
            console.exe
```

Configuration Wizard



Welcome

Choose between creating and extending a domain. Based on your selection,
the Configuration Wizard guides you through the steps to generate a new or extend an existing domain.

● **Create a new WebLogic domain**

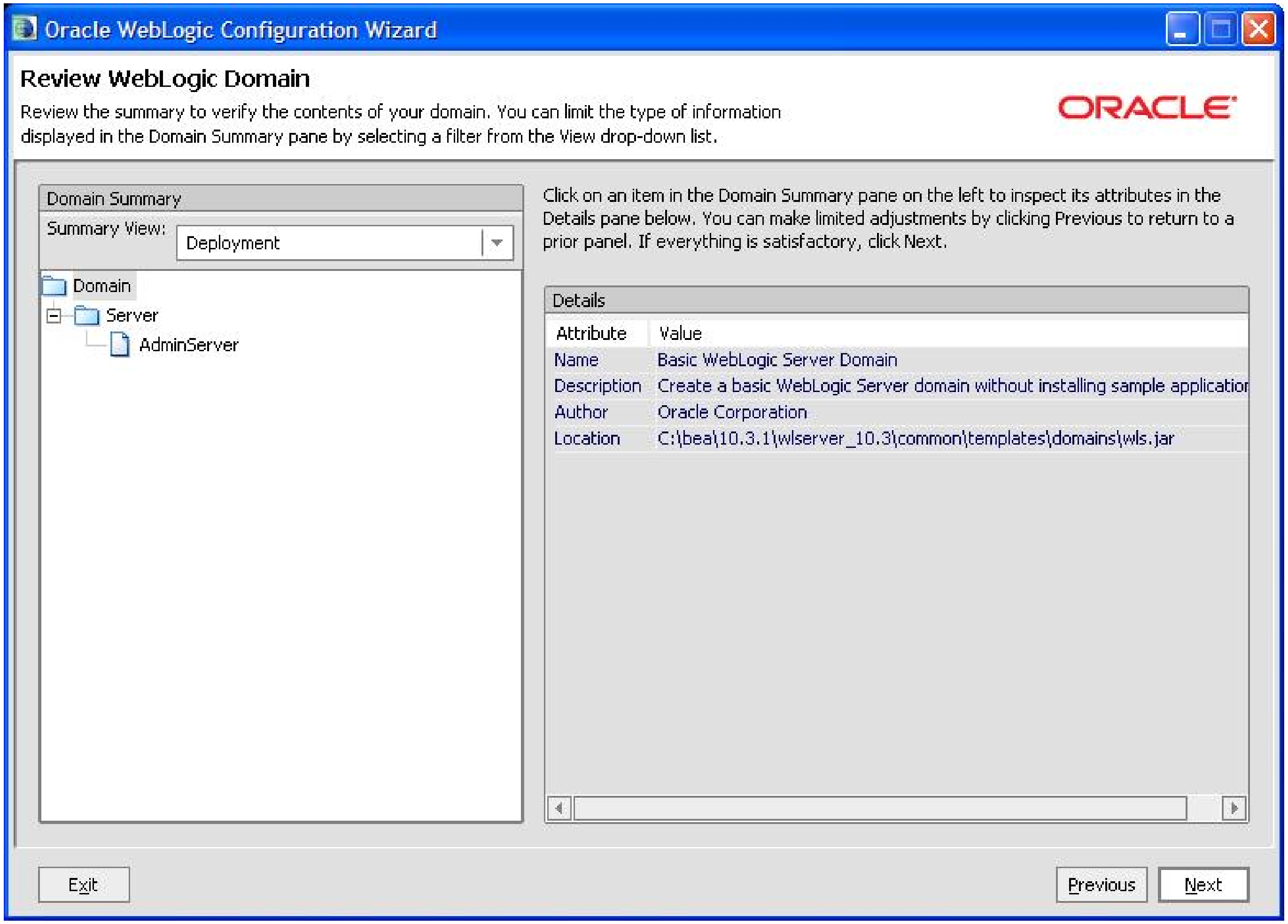
Create a WebLogic domain in your projects directory.

○ **Extend an existing WebLogic domain**

Extend an existing WebLogic domain.

Use this option to add applications and services, or to override existing database access (JDBC) and messaging (JMS) settings.

Configuration Wizard



Review WebLogic Domain

Review the summary to verify the contents of your domain. You can limit the type of information displayed in the Domain Summary pane by selecting a filter from the View drop-down list.

Summary View: Deployment

Click on an item in the Domain Summary pane on the left to inspect its attributes in the Details pane below. You can make limited adjustments by clicking Previous to return to a prior panel. If everything is satisfactory, click Next.

Attribute	Value
Name	Basic WebLogic Server Domain
Description	Create a basic WebLogic Server domain without installing sample applicatio
Author	Oracle Corporation
Location	C:\bea\10.3.1\wlserver_10.3\common\templates\domains\wls.jar

Demo

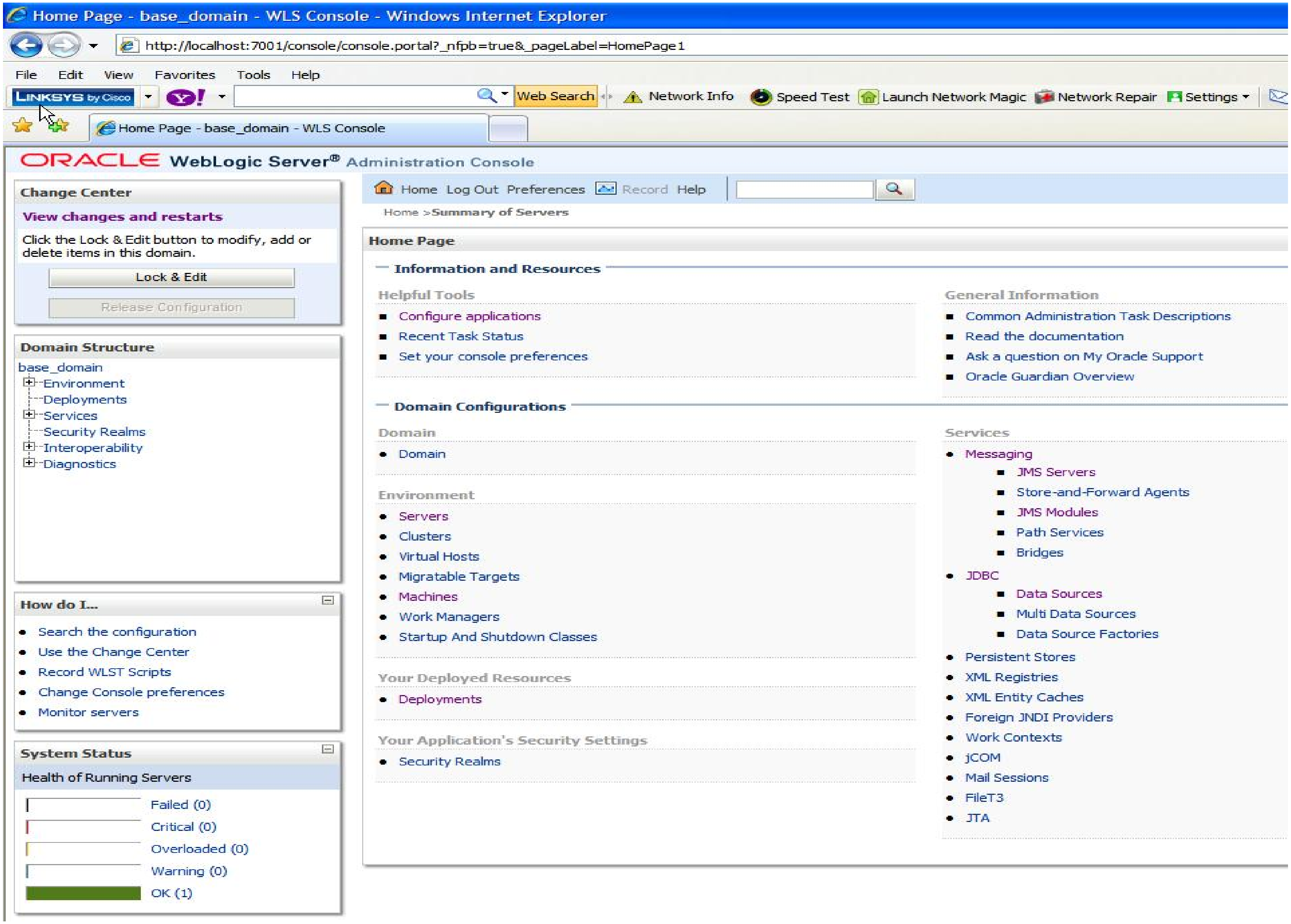
Configuration Wizard

Agenda

- Application server / Weblogic topology
- Download and Installation
- Configuration files. Demo
- Administration Tools: Configuration Wizard. Demo
- Administration Tools: Console. Demo
- Summary
- References
- Q&A

Administration Console

- Is a Web Application deployed on the Admin Server
  - Set of two web applications
    - `console`
    - `consolehelp`
- Deployed on Admin Server
  - http://<AdminServerHost>:<AdminServerPort>/console
  - One-stop shop to manage a WebLogic Domain
    - Configuration Management
    - Services Management
    - Applications Deployment
    - Lifecycle
    - Diagnostics


Home Page - base_domain - WLS Console - Windows Internet Explorer
http://localhost:7001/console/console.portal?_nfpb=true&_pageLabel=HomePage1
File Edit View Favorites Tools Help
LINKSYS by Cisco
Web Search
Network Info
Speed Test
Launch Network Magic
Network Repair
Settings
Home Page - base_domain - WLS Console
ORACLE WebLogic Server® Administration Console
Home Log Out Preferences Record Help

Home >Summary of Servers

Change Center

View changes and restarts

Click the Lock & Edit button to modify, add or delete items in this domain.

Lock & Edit

Release Configuration

Domain Structure

base_domain
- ⊞ Environment
- Deployments
- ⊞ Services
- Security Realms
- ⊞ Interoperability
- ⊞ Diagnostics

How do I...

- Search the configuration
- Use the Change Center
- Record WLST Scripts
- Change Console preferences
- Monitor servers

System Status

Health of Running Servers

- Failed (0)
- Critical (0)
- Overloaded (0)
- Warning (0)
- OK (1)

Home Page

Information and Resources

Helpful Tools

- Configure applications
- Recent Task Status
- Set your console preferences

General Information

- Common Administration Task Descriptions
- Read the documentation
- Ask a question on My Oracle Support
- Oracle Guardian Overview

Domain Configurations

Domain

- Domain

Environment

- Servers
- Clusters
- Virtual Hosts
- Migratable Targets
- Machines
- Work Managers
- Startup And Shutdown Classes

Your Deployed Resources

- Deployments

Your Application's Security Settings

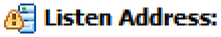
- Security Realms

Services

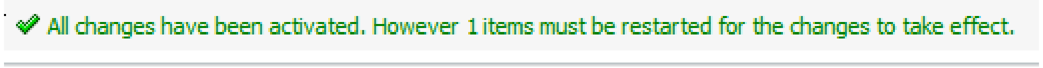
- Messaging
  - JMS Servers
  - Store-and-Forward Agents
  - JMS Modules
  - Path Services
  - Bridges
- JDBC
  - Data Sources
  - Multi Data Sources
  - Data Source Factories
- Persistent Stores
- XML Registries
- XML Entity Caches
- Foreign JNDI Providers
- Work Contexts
- jCOM
- Mail Sessions
- FileT3
- JTA

Change Center

- Provides a way to lock a domain configuration so you can make changes to the configuration (while preventing other accounts from making changes during your edit session)
  - Only one person can change configuration at a given time
- 2 step change process
  - Making the change(s)
  - Activation
    - Changes are propagated to all servers
    - If any server rejects the change, it is reversed from all servers
- 2 step process always enabled for Production domain
  - Disabled by default for Development Domain
    - Changes get automatically activated
- 2 types of changes
  - Dynamic
    - Get activated without server restart
  - Non-Dynamic
    - Require server restart for activation
    - Show up with a specific icon on Admin Console:
      - For example:

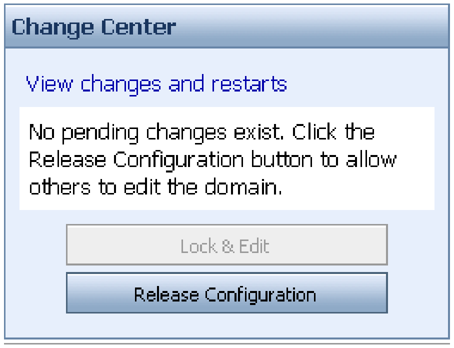


    - An appropriate message shows up in message portlet at the top when you make a change in such a property:



✓ All changes have been activated. However 1 items must be restarted for the changes to take effect.

For Production Domains

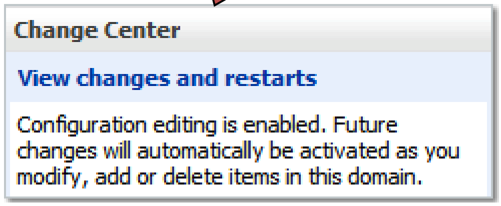


Change Center

View changes and restarts

No pending changes exist. Click the Release Configuration button to allow others to edit the domain.

For Development Domains



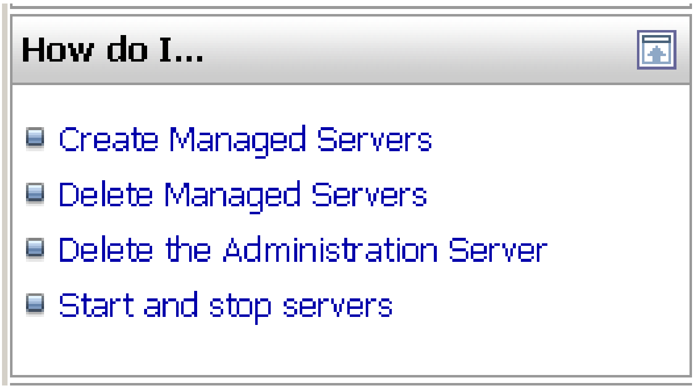
Change Center

View changes and restarts

Configuration editing is enabled. Future changes will automatically be activated as you modify, add or delete items in this domain.

“How do I…”

- Provides Context Sensitive Help
- Links to associated documentation for a console page
- Serviced by “consolehelp” web application



How do I...

- Create Managed Servers
- Delete Managed Servers
- Delete the Administration Server
- Start and stop servers

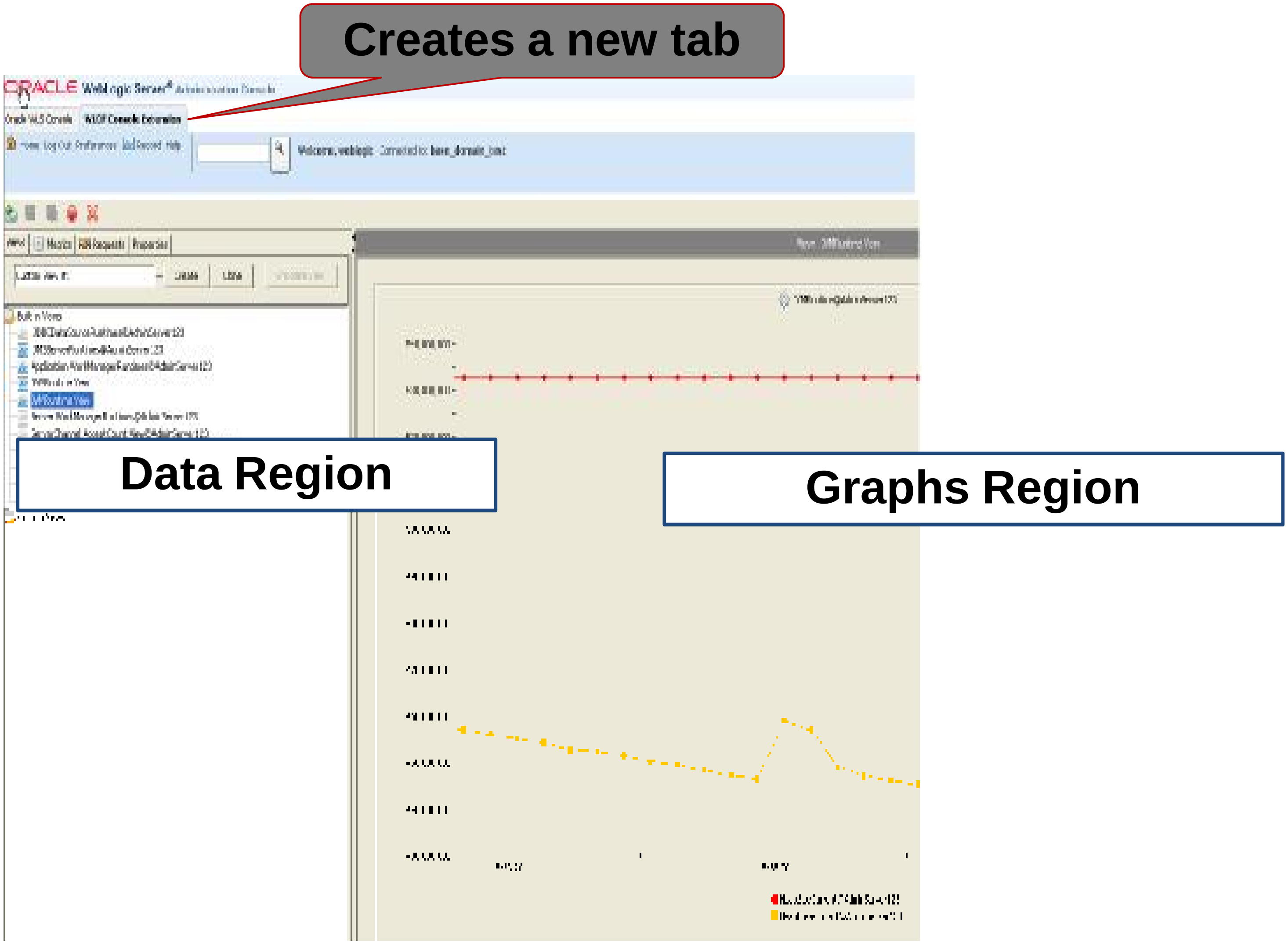
WLDF Console Extension

- Provides diagnostic data about all servers in the domain
  - For instance
    - Memory utilization
      - Garbage collection behavior
    - JVM Uptime
- Documentation
  - http://edocs.bea.com/wls/docs103/pdf/wldf_console_ext.pdf
- Provides data over and above what is available in the Admin console
- Allows creation of custom graphs

How to add WLDF Console Extension?

- Find the `diagnostics-console-extension.jar` file in the `WL_HOME\server\lib\console-ext` directory
  - where `WL_HOME` is the directory in which you installed WebLogic Server.
- Copy `diagnostics-console-extension.jar` into the `DOMAIN-DIR/console-ext` directory for every domain in which you want to use the Console Extension
  - where `DOMAIN-DIR` is the domain's root directory.
- Restart the Administration Server for the domain where you installed the extension
  - WLDF Console Extension will show up as a new adjacent tab when you access the Admin console

WLDF Console Extension


ORACLE WebLogic Server Administration Console
Oracle WLS Console
WLDF Console Extension
Home Log Out Preferences Record Help
Welcome, weblogic Connected to: base_domain_bmc
Views Metrics Requests Properties
Built-in Views
JVMRuntimeView

Creates a new tab

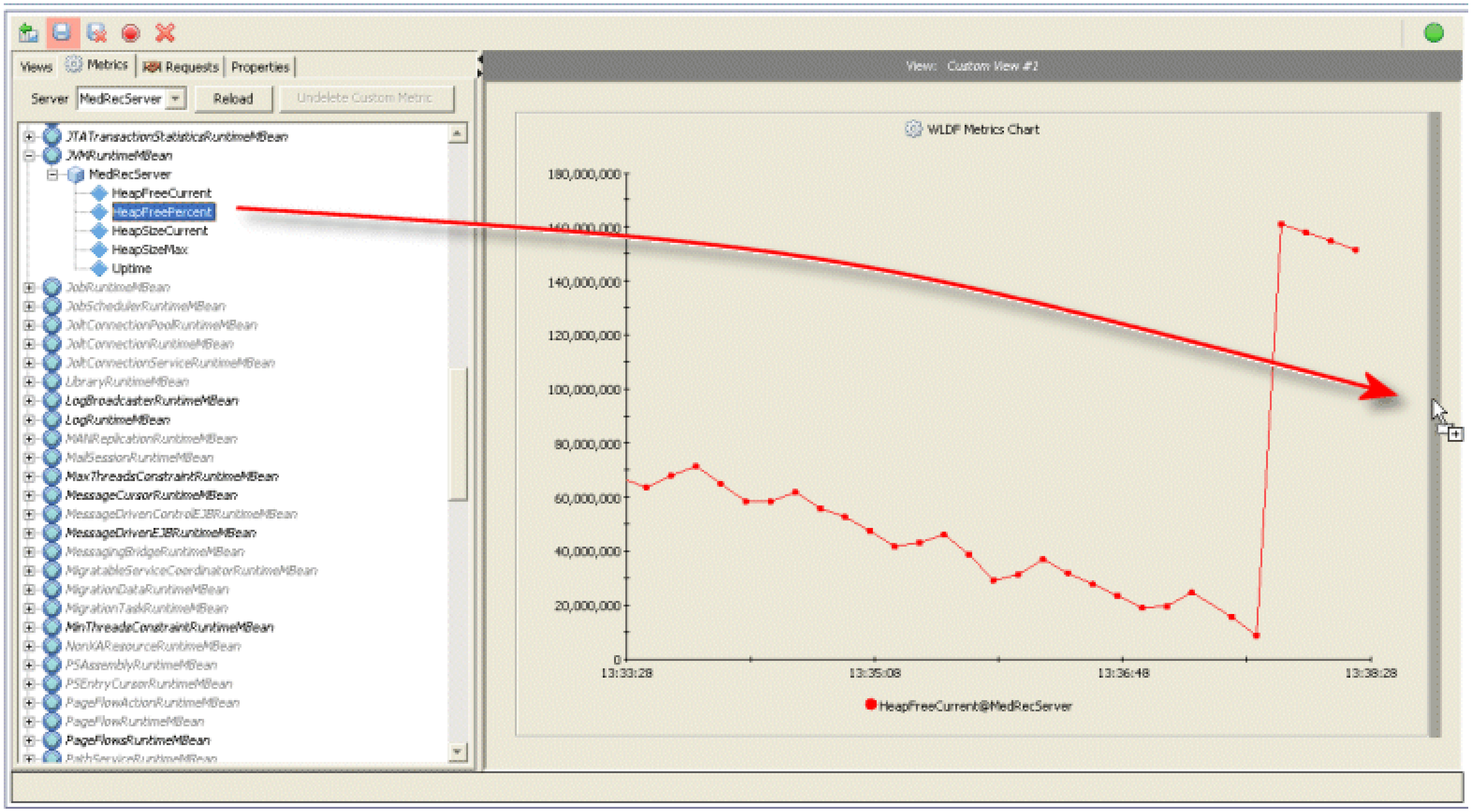
Data Region

Graphs Region

WLDF Console E

Creating Customized

Drag and drop any attribute from “Metrics” tab to a new or existing graph in the View


Views
Metrics
Requests
Properties
Server MedRecServer
Reload
Undelete Custom Metric
JTATransactionStatisticsRuntimeMBean
JVMRuntimeMBean
MedRecServer
HeapFreeCurrent
HeapFreePercent
HeapSizeCurrent
HeapSizeMax
Uptime
JobRuntimeMBean
JobSchedulerRuntimeMBean
JoltConnectionPoolRuntimeMBean
JoltConnectionRuntimeMBean
JoltConnectionServiceRuntimeMBean
LibraryRuntimeMBean
LogBroadcasterRuntimeMBean
LogRuntimeMBean
MANReplicationRuntimeMBean
MailSessionRuntimeMBean
MaxThreadsConstraintRuntimeMBean
MessageCursorRuntimeMBean
MessageDrivenControlEJBRuntimeMBean
MessageDrivenEJBRuntimeMBean
MessagingBridgeRuntimeMBean
MigratableServiceCoordinatorRuntimeMBean
MigrationDataRuntimeMBean
MigrationTaskRuntimeMBean
MinThreadsConstraintRuntimeMBean
NonXAResourceRuntimeMBean
PSAssemblyRuntimeMBean
PSEntryCursorRuntimeMBean
PageFlowActionRuntimeMBean
PageFlowRuntimeMBean
PageFlowsRuntimeMBean
PathServiceRuntimeMBean
View: Custom View #1
WLDF Metrics Chart
180,000,000
160,000,000
140,000,000
120,000,000
100,000,000
80,000,000
60,000,000
40,000,000
20,000,000
0
13:33:28
13:35:08
13:36:48
13:38:28
● HeapFreeCurrent@MedRecServer

Demo

Admin Console

- Tool Bar: Identifies domain, user

Links to common features such as setting Preferences, Logging out

Always present

-Breadcrumb Navigation: Shows Trail of links to artifacts visited

- System Status: Summary of current health of all servers in the domain

Clicking on the status takes you to the definition of all servers in that status

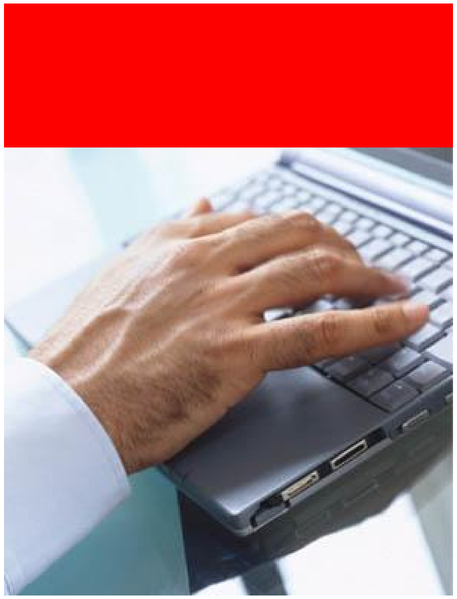
Updated with each page refresh

- How do I: Provides Context Sensitive Help

- 2 modes: Production and Development

- Change Listen Port Field and Save

SUMMARY



- Weblogic server topology: Domain, Administration Server, Managed Server, Cluster, Node Manager
- How to install WebLogic Server
- How to start/stop WebLogic Server
- Configuration Wizard
- Administration Console and Console Extension

References

- WebLogic Server 10.3 Documentation

  http://download.oracle.com/docs/cd/E12840_01/wls/docs103/sitemap.html
- Tutorials on WebLogic Server on Oracle Technology Network

  http://www.oracle.com/technology/obe/fusion_middleware/index.htm

- Note 733716.1 - Getting Started with Oracle WebLogic Server: A Guide For OracleAS Administrators
- OU Classes
  - Oracle WebLogic Server 11g: Administration Essentials (D58682GC10)
  - Oracle WebLogic Server 11g: Advanced Administration (D58686GC10)

Q&A