Proven Practice

Application Servers - BEA WebLogic Configuration and IBM Cognos Application Deployment

Product(s): IBM Cognos 8.4, BEA WebLogic

Area of Interest: Infrastructure

DOC ID: AS11 Version 8.4.0.0

Copyright and Trademarks

This document is maintained by the Best Practices, Product and Technology team. You can send comments, suggestions, and additions to cscogpp@ca.ibm.com.

TABLE OF CONTENT

1 Introduction

1.1 Purpose

This document is for those who have limited experience with WebLogic and who wish to set up IBM Cognos 8 BI on a WebLogic Server environment. The primary goal is to set up a basic environment for testing purposes and not necessarily a high performance production environment.

1.2 Scope

The instructions in this document will work cross platform however the examples shown will be based on a Windows install. UNIX installations require slight modifications to syntax but should otherwise be identical. For example, regarding the syntax for variables, use ${VARIABLE} for UNIX and %VARIABLE% for Windows. Consult your UNIX documentation for more information regarding commands, syntax and scripting.

WebLogic is designed so that only one WebLogic Server instance in a domain can run the Administration Service. This server is called the Administration Server. All other servers in the domain are managed by the Administration Server and obtain their runtime configuration from it. These servers are called Managed Servers. IBM Cognos 8 BI is an application that will be deployed to a Managed Server.

The Administration Server is the single point of control for the entire domain, and it must be running for you to make changes to the configuration of any of the Managed Servers running beneath it.

This document will take you through the steps to configure WebLogic Server and deploy Cognos 8.

1.3 Definitions, Acronyms, and Abbreviations

Term	Definition
JVM	Java Virtual Machine A software "execution engine" that safely and compatibly executes the byte codes in Java class files on a microprocessor (whether in a computer or in another electronic device). The JVM is responsible for the hardware and operating system independence of the J2SE platform, the small size of compiled code (bytecodes), and platform security.
J2SE	Java 2 Platform, Standard Edition There are two principal products in the J2SE platform family: J2SE Runtime Environment (JRE) and J2SE Development Kit (JDK).
JRE	Java Runtime Environment The JRE provides the Java APIs, Java virtual machine, and other components necessary to run applets and applications written in the Java programming language. It is also the foundation for the technologies in the Java 2 Platform, Enterprise Edition (J2EE) for enterprise software development and deployment. The JRE does not contain tools and utilities such as compilers or debuggers for developing applets and applications
JDK	Java Development Kit The JDK is a superset of the JRE, and contains everything that is in

	the JRE, plus tools such as the compilers and debuggers necessary for developing applets and applications.
J2EE	Java 2 Platform, Enterprise Edition Combines a number of technologies in one architecture with a comprehensive Application Programming Model and Compatibility Test Suite for building enterprise-class server-side applications. The Java 2 Platform, Enterprise Edition (J2EE) defines the standard for developing multitier enterprise applications. The J2EE platform simplifies enterprise applications by basing them on standardized, modular components, by providing a complete set of services to those components, and by handling many details of application behavior automatically, without complex programming.
WL	BEA WebLogic Server
WL_HOME	Path to the directory where WebLogic is installed (example e:\bea\)
WLS	WebLogic Server
WebLogic Domain	A domain is a collection of resources, such as servers, clusters, database connections, security services, and J2EE applications, that you manage as a unit. A domain must contain at least a definition for an Administration Server and an administrative user. You can define additional resources as part of the domain-creation process or at any time after creating the domain.
Administration Server	The Administration Server is the WebLogic Server used to configure and manage all the WebLogic Servers in its domain. A domain may include multiple WebLogic Server clusters and independent WebLogic Server instances. If a domain contains only one WebLogic Server, then that server is the Administration Server. In a domain with multiple instances of WebLogic Server, the first instance to start must be the Administration Server. For a typical production system, BEA recommends that you deploy your applications only on Managed Servers. This practice allows you to dedicate the Administration Server to configuration and monitoring of the domain, while one or more Managed Servers service your applications.
Managed Server	The Managed Server is a WebLogic Server instance that retrieves its configuration data from the domain's Administration Server. There can be many Managed Servers in a domain, but only one Administration Server. Usually, you create and start server instances as Managed Servers to run your business applications in a production environment. In this standard scenario, the server instance that you start as the Administration Server does not run business applications. Instead, it only manages resources in the domain.
Administration Console	The WebLogic Server Administration Console runs in a Web browser. It displays the components of the domain it administers, including clusters and independent WebLogic Servers, in a graphical tree in the left pane. The right pane displays details about the object selected in the left pane.
Node Manager	Node Manager is a Java utility that runs as separate process from WebLogic Server and allows you to perform common operations tasks for a Managed Server, regardless of its location with respect to its Administration Server. Use of Node Manager is optional.
SSL	Short for Secure Sockets Layer, a protocol developed by Netscape for transmitting private documents via the Internet. SSL uses a

	cryptographic system that uses two keys to encrypt data − a public key known to everyone and a private or secret key known only to the recipient of the message. By convention, URLs that require an SSL connection start with HTTPS instead of HTTP.
keytool	Key and Certificate Management Tool It enables users to administer their own public/private key pairs and associated certificates for use in self-authentication (where the user authenticates himself/herself to other users/services) or data integrity and authentication services, using digital signatures. It also allows users to cache the public keys (in the form of certificates) of their communicating peers.
Keystore	A keystore is a protected database that holds keys and certificates for an enterprise. Access to a keystore is guarded by a password (defined at the time the keystore is created, by the person who creates the keystore, and changeable only when providing the current password). In addition, each private key in a keystore can be guarded by its own password.

1.4 Assumptions

Before continuing, make sure the following prerequisites have been met.

- That you have already installed WebLogic Server and have followed the steps outlined in Application Servers - BEA WebLogic Server: Installing the Application Server.
- That you have already installed IBM Cognos 8 and have followed the steps outlined in Application Servers - Installing and Configuring the IBM Cognos Application.
- That you have the necessary access rights to run WebLogic Server and to deploy IBM Cognos 8. It is best if the user used to install WebLogic Server is the same user used to setup and run IBM Cognos 8. You also need the userID and password required to access the WebLogic Administration Console.
- That you have the necessary system resources to perform the installation. There is considerably more memory, hard disk space and CPU power required to run any full Application Server than running Tomcat. The minimal requirements for WebLogic should be compared to available resources. Running with less than 768MB of RAM for the Max Heap Memory is not suggested (although it may be possible for testing purposes only), 1GB should be considered a minimum to ensure performance and reliability.

  For WebLgoic 8.1.x, you should allot 1 GB of disk space for the application server installation. More space will be required for added Managed Servers and application files, logging and additional domains.
- The JVM used is the one shipped with WebLogic or one that is supported by WebLogic and IBM Cognos 8. If you are using version 8.1.x, then you can use the JVM shipped with WebLogic or an existing JDK 1.4.2 on the system. It is recommended that you use the JDK shipped with WebLogic, though.
- The WebLogic Server install for AIX does not provide a JVM.

- You must install this before installing WebLogic. Note: IBM Cognos 8 does not currently support JRockit.

- That you have the customer documentation available as a resource. You should have both the IBM Cognos 8 documentation as well as the BEA WebLogic documentation.

- That you are familiar with the operating system (OS) with which you are using. This document assumes you are familiar with the OS. This includes user commands at the command line, syntax and scripting.

- That you are familiar with IBM Cognos 8. You should already know how to setup and run the product using Tomcat.

- That you are familiar with SSL and security concepts. The WebLogic demonstration keystores are used as examples in this document. The user can obtain certificates from the appropriate sources. (For example, Verisign, Thawte, etc.) They are not provided by IBM Cognos Support.

2 Preparation for Cognos Application Deployment

Various configuration requirements must be met before running IBM Cognos 8 on any machine.

Steps general to all application servers are described in the Application Servers – Installing and Configuring the IBM Cognos Application document. You will need to set up the database clients that you intend to use with IBM Cognos 8. You should also consult your database documentation and the IBM Cognos 8 Installation and Configuration Guide included with the IBM Cognos 8 install for more information.

2.1 Setting up the Environment

For most simple installations, it is recommended that environment variables values be set globally. For Windows, these would be system environment variables. On UNIX, it could be a script to be sourced. WebLogic installs a script called setEnv.cmd (Windows) or setEnv.sh (UNIX) in the domain directory that can be altered to include the appropriate variable/value pairs. For UNIX, users should be sure to set the variables correctly according to their default shell. Different platforms and different shell environments use slightly different syntax to accomplish this task. You should consult your UNIX documentation for more information.

2.1.1 Steps to Modify the setEnv Script

The setEnv script can be used to set environment settings for the domain. Its use is optional.

Important: You must add a command to call the setEnv script when a server instance is started, otherwise it will be ignored.

Set environment variables such as JAVA_HOME, PATH or library path and your database environment variables. Variable names differ depending on the OS platform and the database vendor. You must append the <c8_install_location>/bin directory to the library path so that the Cognos library files can be accessed.

If you intend to use a single server install, you can set the PATH or library path environment variable, here. If you are installing separate components in different locations, you should set the PATH or library path for each Managed Server instance to its corresponding Cognos install.

2.1.1.1 WebLogic 8.1.x

You will need to add the same variable settings described in the "Steps to Modify the WebLogic 8.1.x Environment" section.

- set JAVA_OPTIONS
- set JAVA_VM
- set PRODUCTION_MODE
- set MEM_ARGS
- set library path

These variables are described later in this document.

Sample setEnv.sh script for HPUX

```
#!/bin/sh
# ****************************************************************************
# This script is used to set up your environment for development with
# WebLogic Server. It simply calls the commEnv.sh script under
# /system_1/weblogic/8.1.3/weblogic81/common/bin.  Add domain specific
# configuration in this script below.
# ****************************************************************************

# set up WL_HOME, the root directory of your WebLogic installation

WL_HOME="/system_1/weblogic/8.1.3/weblogic81"

# Set Production Mode.  When this is set to true, the server starts up in

# production mode.  When set to false, the server starts up in development

# mode.  If it is not set, it will default to false.

PRODUCTION_MODE=”true”

# Set JAVA_VENDOR to java virtual machine you want to run on server side.

JAVA_VENDOR="HP"

# Set JAVA_HOME to java virtual machine you want to run on server side.

JAVA_HOME="/system_1/weblogic/8.1.3/jdk142_03"

# set up common environment

PATH=”${PATH}:/system_1/weblogic/8.1.3/cognos/c8/bin”

JAVA_VM=”-server”

JAVA_OPTIONS=”-Dweblogic.security.SSL.trustedCAKeyStore=${WL_HOME}/server/lib/cacerts -
Dorg.xml.sax.driver=org.apache.xerces.parsers.SAXParser -
Dweblogic.management.discover=true”

MEM_ARGS=”-Xms768m -Xmx768m”

# set up database environment

# This is an example for Oracle

ORACLE_HOME=”/oracle/9.2.0.5”

TNS_ADMIN=”/system_1/weblogic/config”

NLS_LANG=”AMERICAN_AMERICA.UTF8”

PATH=”${PATH}:${ORACLE_HOME}/bin:/system_1/weblogic/8.1.3/jdk142_03/bin:/public/tools/Acr
obat5/bin”

SHLIB_PATH=”${ORACLE_HOME}/lib32:/system_1/weblogic/8.1.3/cognos/c8/bin:${SHLIB_PATH}”

. "${WL_HOME}/common/bin/commEnv.sh"

CLASSPATH="${WEBLOGIC_CLASSPATH}${CLASSPATHSEP}${POINTBASE_CLASSPATH}${CLASSPATHSEP}${JAV
A_HOME}/jre/lib/rt.jar${CLASSPATHSEP}${WL_HOME}/server/lib/webservices.jar"

export CLASSPATH
```

2.2 Steps to Modify the WebLogic 8.1.x Environment

For an Administration Server and Managed Server configuration, you will modify the startManagedWebLogic file. (*.cmd files are for Windows, *.sh files are for UNIX platforms.) These files are located in the directory for the WebLogic domain you created using the Domain Configuration Wizard. For example, if you installed WebLogic in C:\bea8 and created a domain called cognos, you would find these files in C:\bea8\user_projects\domains\cognos.

Note: For the UNIX platforms, when setting the variable values, you'll have to enclose the strings in quotes. This is not necessary for Windows.

1. Make sure you set the server name. For example, if you created a Managed Server called server1, set the SERVER_NAME to server1.

   ```
   set SERVER_NAME=server1
   ```

   Note: The syntax used for UNIX is different. The set command is not used. Instead, use a variable assignment.

2. Append <cognos8_location>\bin to the appropriate environment variable.

Windows:	PATH
AIX:	LIBPATH
Solaris:	LD_LIBRARY_PATH
HPUX:	SHLIB_PATH

   You can set this in the script or as part of your environment.

3. Set the JVM type if the JVM supports different modes. The JVM mode is a text string that indicates the mode in which you want the JVM to run. The values that you supply depend on the JVM that you are using. For example, the Sun JDK can run a -hotspot, -client or -server JVM. The default value is -hotspot. Change the default setting from

   ```
   JAVA_VM=
   ```

   to

   ```
   JAVA_VM=-server
   ```

   The IBM JRE does not support the –server option so no changes are required.

4. Set the Java options by appending -Dorg.xml.sax.driver=org.apache.xerces.parsers.SAXParser and -Dweblogic.management.discover=false to the Java environment. The java options are one or more Java arguments that you want to pass to the JVM.

   ```
   set JAVA_OPTIONS=-
   Dweblogic.security.SSL.trustedCAKeyStore=d:\bea\weblogic813\server
   \lib\cacerts -
   Dorg.xml.sax.driver=org.apache.xerces.parsers.SAXParser -
   Dweblogic.management.discover=false
   ```

   Note: Forward slashes are used for UNIX.

5. You can start your domain in one of two modes, development or production. You use development mode while you are developing your applications. Development mode uses a relaxed security configuration. You use production mode when your application is running in its final form. A production domain uses full security and may use clusters or other advanced features.

   Change the PRODUCTION_MODE variable to true to enable production mode. If the variable isn't set, the default is Development mode.

   ```
   PRODUCTION_MODE=
   ```

   to

   ```
   PRODUCTION_MODE =true
   ```

6. Specify the minimum and maximum values (in megabytes) for Java heap memory. For example,

   ```
   set MEM_ARGS=-Xms768m –Xmx768m
   ```

   BEA recommends that the minimum and maximum values be the same so that the JVM does not resize the heap. The values assigned to these parameters can dramatically affect the performance of your WebLogic Server and are provided here only as general defaults.

In a production environment you should carefully consider the correct memory heap size to use for your applications and environment. The settings depend on the amount of memory available on your machine. For example, if you have a machine which has more than 2 GB of memory, a minimum of 1024m and a maximum of 1024m with the young generation values set to 256m are suggested starting values that you can adjust for your environment.

```
set MEM_ARGS=-XX:NewSize=256m -XX:MaxNewSize=256m -Xms1024m -
Xmx1024m
```

7. Save and close the file.

Example of the startManagedWebLogic.cmd

```
@rem ***************************************************************************
@rem This script is used to start a managed WebLogic Server for the domain in
@rem the current working directory.  This script reads in the SERVER_NAME and
@rem ADMIN_URL as positional parameters, sets the SERVER_NAME variable, then
@rem starts the server.
@rem
@rem Other variables that startWLS takes are:
@rem
@rem WLS_USER       - cleartext user for server startup
@rem WLS_PW         - cleartext password for server startup
@rem PRODUCTION_MODE      - Set to true for production mode servers, false for
@rem                  development mode
@rem JAVA_OPTIONS   - Java command-line options for running the server. (These
@rem                  will be tagged on to the end of the JAVA_VM and MEM_ARGS)
@rem JAVA_VM        - The java arg specifying the VM to run.  (i.e. -server,
@rem                  -hotspot, etc.)
@rem MEM_ARGS       - The variable to override the standard memory arguments
@rem                  passed to java
@rem
@rem For additional information, refer to the WebLogic Server Administration
@rem Guide (http://e-docs.bea.com/wls/docs81/ConsoleHelp/startstop.html).
@rem ***************************************************************************

echo off

SETLOCAL

set WL_HOME=D:\bea813\weblogic81

@rem Set Production Mode.  When this is set to true, the server starts up in

@rem production mode.  When set to false, the server starts up in development

@rem mode.  If it is not set, it will default to false.

set PRODUCTION_MODE=true

@rem Set JAVA_VENDOR to java virtual machine you want to run on server side.

set JAVA_VENDOR=Sun

@rem Set JAVA_HOME to java virtual machine you want to run on server side.

set JAVA_HOME=D:\bea813\jdk142_04

call "%WL_HOME%\common\bin\commEnv.cmd"

@rem Set SERVER_NAME to the name of the server you wish to start up.

set ADMIN_URL=http://localhost:7001

set SERVER_NAME=server1

@rem Set WLS_USER equal to your system username and WLS_PW equal

@rem to your system password for no username and password prompt

@rem during server startup.  Both are required to bypass the startup

@rem prompt.
```

```
set WLS_USER=

set WLS_PW=

@rem Set JAVA_VM to java virtual machine you want to run on server side.

set JAVA_VM=-server

@rem Set JAVA_OPTIONS to the java flags you want to pass to the vm. i.e.:

@rem set JAVA_OPTIONS=-Dweblogic.attribute=value -Djava.attribute=value

set JAVA_OPTIONS=-Dweblogic.security.SSL.trustedCAKeyStore=%WL_HOME%\server\lib\cacerts -
Dorg.xml.sax.driver=org.apache.xerces.parsers.SAXParser -
Dweblogic.management.discover=false

@rem Set MEM_ARGS to the memory args you want to pass to java.  For instance:

@rem if "%JAVA_VENDOR%"=="BEA" set MEM_ARGS=-Xms32m -Xmx200m

set MEM_ARGS=-Xms768m -Xmx768m

@rem Set SERVER_NAME and ADMIN_URL, they must by specified before starting

@rem a managed server, detailed information can be found at

@rem http://e-docs.bea.com/wls/docs81/adminguide/startstop.html.

if "%1" == ""  goto checkEnvVars

set SERVER_NAME=%1

if "%2" == "" goto checkEnvVars

set ADMIN_URL=%2

goto callWebLogic

:checkEnvVars

if "%SERVER_NAME%" == "" goto usage

if "%ADMIN_URL%" == ""  goto usage

set SERVER_NAME="%SERVER_NAME%"

set ADMIN_URL="%ADMIN_URL%"

goto callWebLogic

:usage

echo Need to set SERVER_NAME and ADMIN_URL environment variables or specify

echo them in command line:

echo Usage: startManagedWebLogic [SERVER_NAME] [ADMIN_URL]

echo for example:

echo startManagedWebLogic managedserver1 http://localhost:7001

goto finish

:callWebLogic

@rem Start WebLogic Server

set
CLASSPATH=%WEBLOGIC_CLASSPATH%;%POINTBASE_CLASSPATH%;%JAVA_HOME%\jre\lib\rt.jar;%WL_HOME%
\server\lib\webservices.jar;%CLASSPATH%

@echo.

@echo CLASSPATH=%CLASSPATH%

@echo.

@echo PATH=%PATH%

@echo.
```

```
@echo ************************************************
@echo *  To start WebLogic Server, use a username and   *
@echo *  password assigned to an admin-level user.  For *
@echo *  server administration, use the WebLogic Server *
@echo *  console at http://[hostname]:[port]/console    *
@echo ************************************************

"%JAVA_HOME%\bin\java" %JAVA_VM% %MEM_ARGS% %JAVA_OPTIONS% -Dweblogic.Name=%SERVER_NAME%
-Dweblogic.management.username=%WLS_USER% -Dweblogic.management.password=%WLS_PW% -
Dweblogic.management.server=%ADMIN_URL% -
Djava.security.policy="%WL_HOME%\server\lib\weblogic.policy" weblogic.Server

:finish

ENDLOCAL
```

2.3 Steps to Modify the WebLogic 9.x Environment

The startup scripts are located in %WL_HOME%\user_projects\domains\<*domain_name*>\bin. For example, if WebLogic Server is installed in D:\bea\9.0.0, and the domain name is cognos8, then the startup scripts would be located in D:\bea\9.0.0\user_projects\domains\cognos8\bin.

2.3.1 Serial Version UID

A serial version UID mismatch is encountered when using Weblogic Server 9.x with JDK 5.0 (version 1.5.0_03 or later). This affects all platforms. Before running WebLogic Server, implement the following workaround.

1. Navigate to $BEA_HOME/weblogic9x/common/bin and locate the commEnv.sh file.
2. Open commEnv.sh in an editor and modify it to include the following command:

   ```
   JAVA_OPTIONS="${JAVA_OPTIONS}  -
   Dcom.sun.xml.namespace.QName.useCompatibleSerialVersionUID=1.0"

   export JAVA_OPTIONS
   ```

3. Save and close the file.

2.3.2 Configuring the WebLogic 9.x Environment

The steps to modify the startWeblogic.cmd or startManagedWeblogic.sh script are similar to those for WebLogic Server 8.1.x. However, the recommended way to configure WebLogic Server 9.x is to use the Administration Server UI in conjunction with Node Manager. This allows you remotely start and stop the managed servers via the Administration Console, as well as manage resources.

NOTE: For WLS9.2 on AIX if you encounter core and java cores, use the WLS scripts to start the Managed Node and NOT the Administration Console.

1. Since there is no way to set environment variables in the Administration Console, you will need to set your environment prior to starting WebLogic Server processes. You should set environment variables such as PATH or library path and your database environment variables. Variable names differ depending on the OS platform and the database vendor. You must append the <c8_install_location>/bin directory to the library path so that the Cognos library files can be accessed.

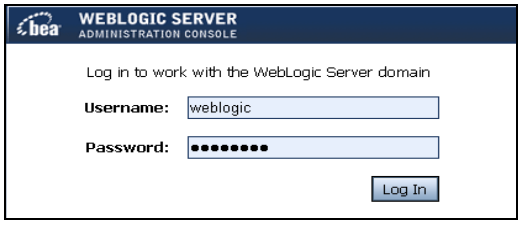
Windows:	PATH
AIX:	LIBPATH
Solaris or Linux:	LD_LIBRARY_PATH
HPUX:	SHLIB_PATH

You can set this in a separate script or as part of your environment.

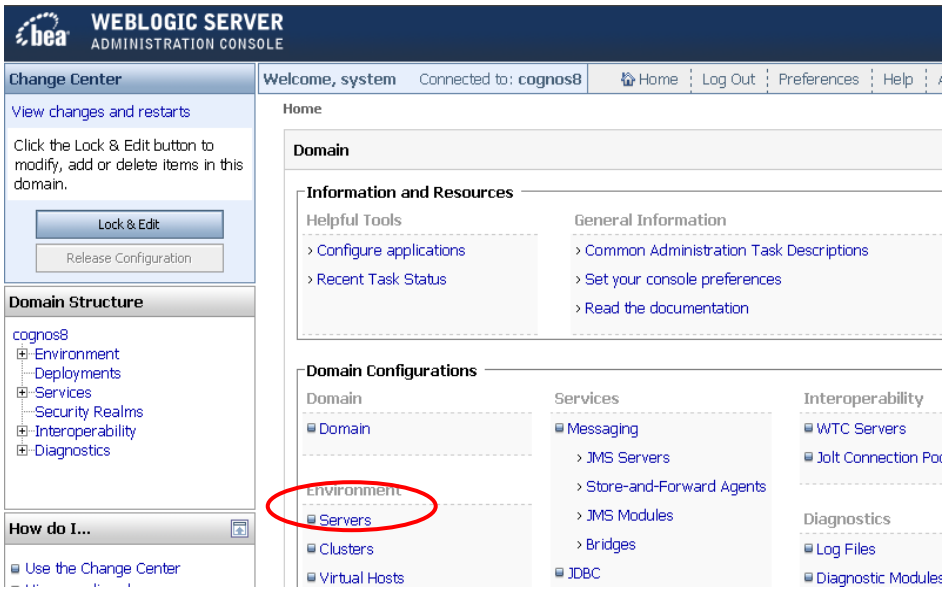
2. Node Manager must be started in order to start and stop Managed Server instances via the Administration Console. In Windows, the Node Manager can be started as a service, via the Windows Services Console. In UNIX, it can be started from ${WL_HOME}/weblogic90/server/bin by running the startNodeManager.sh script. The default port used by Node Manager is 5556. There are no configuration steps required for Node Manager in order to run IBM Cognos 8.

3. The Administration Server must be started prior to accessing the Administration Console. To start the Administration Server, run startWebLogic.cmd (Windows) or startWebLogic.sh (UNIX) from %WL_HOME%\user_projects\domains\<*domain_name*>\bin.

4. To access the Administration Console, open a browser session and connect to http://<*host_name*>:<*port*>/console. For example, if WebLogic Server is installed on a machine called "server1" using port number 7001 for the Administration Server, then you would access http://server1:7001/console. When the login page appears, enter the username and password used when creating the administrative user in the Domain Configuration Wizard.

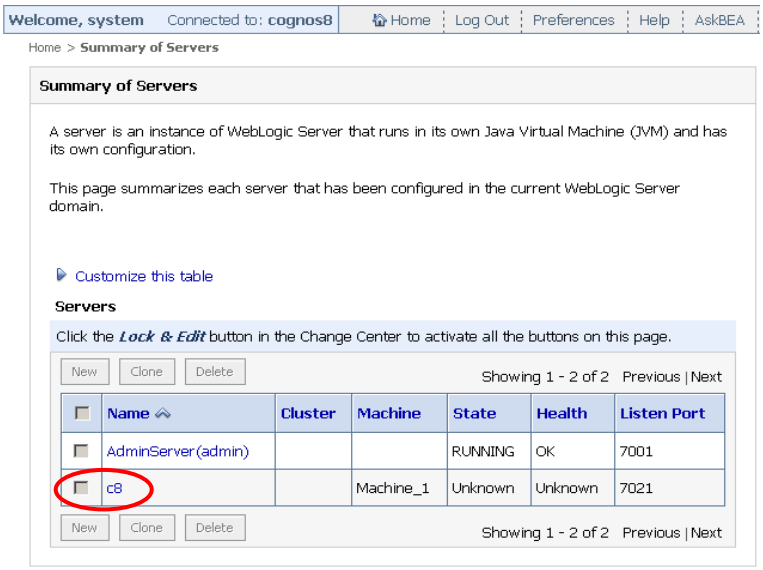


4. Once authenticated, you will see the Administration Console UI. The following view is the Home page. Click on the Servers link to view the Managed Server instances created for the domain.



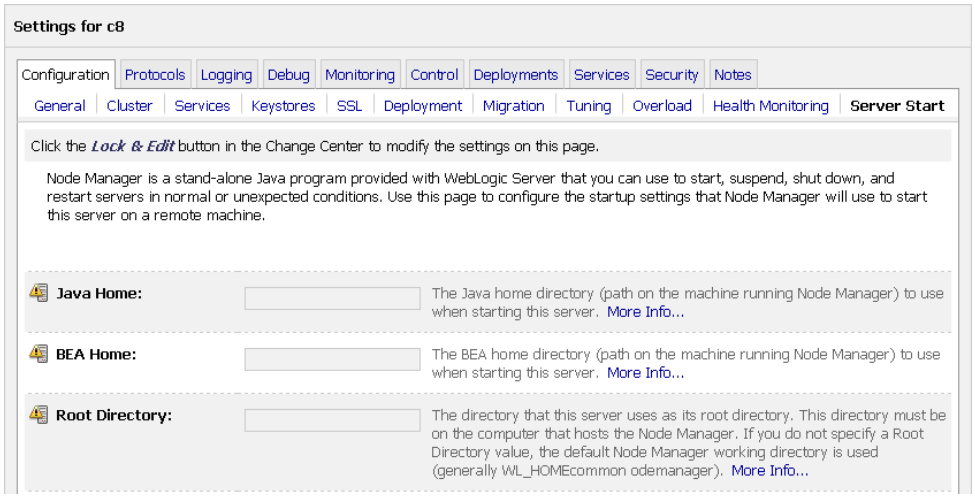
5. You will see the Summary of Servers for this domain. The server instance operating as the Administration Server is displayed with the '(admin)' label after its name. Click on the Managed

Server instance to begin its configuration. In this example, we'll be using the Managed Server called c8.



6. Under the Configuration tab, select the Server Start tab.

The page will display the startup parameters that can be set for each server instance.



7. In order to edit the page, you must click the Lock & Edit button.

Once in Edit mode, enter the Java Home value and the Java Arguments.

The Java Home path must be the same JAVA_HOME used for IBM Cognos 8. It is recommended to use the JVM included with the WebLogic install.

The Java Arguments include the memory settings, JVM type and the org.xml.sax.driver setting (-Dorg.xml.sax.driver=org.apache.xerces.parsers.SAXParser).

For memory, you need to specify the minimum and maximum values (in megabytes) for Java heap memory. For example, -Xms768m –Xmx768m. Also, you need to set the MaxPermSize.

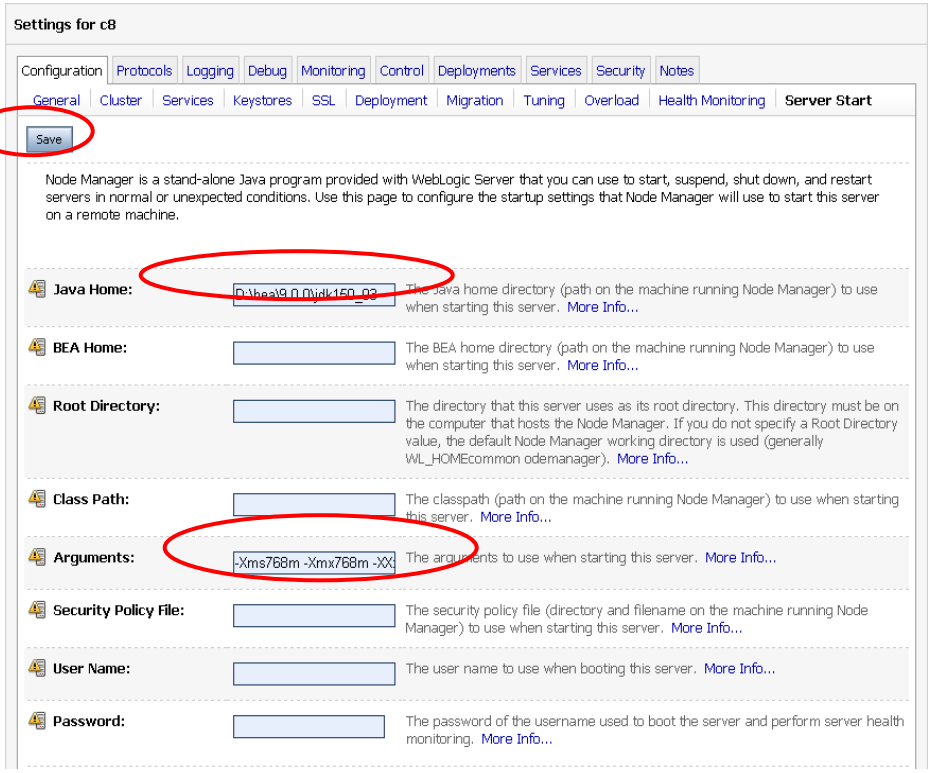
BEA recommends that the minimum and maximum values be the same so that the JVM does not resize the heap. Also, you can set the MaxPermSize to a quarter of the heap size. The values assigned to these parameters can dramatically affect the performance of your WebLogic Server and are provided here only as an example.

```
-Xms768m -Xmx768m -XX:MaxPermSize=128m -server -
Dorg.xml.sax.driver=org.apache.xerces.parsers.SAXParser
```

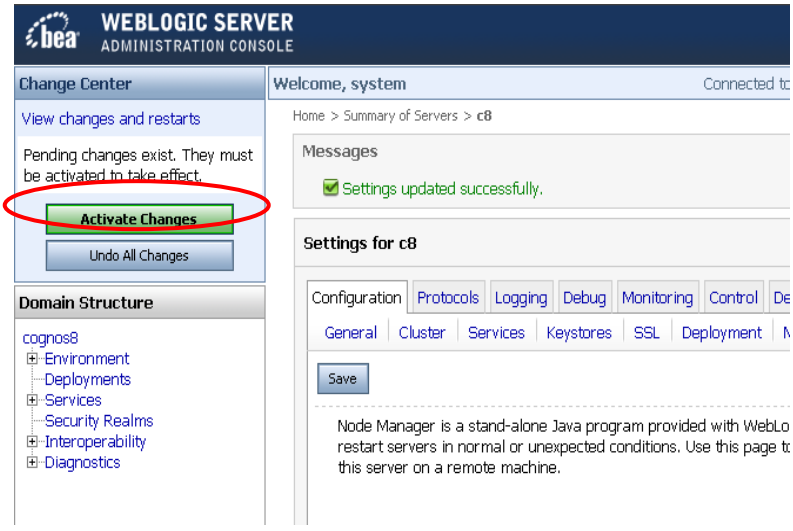
You can also enter values for the other properties if you do not wish to use the defaults. For example, you wish to use a different WebLogic userID to start the Managed Server.

Note: The IBM JRE does not support the –server option. If WebLogic is running on an IBM JRE, do not add the –server option.

Click the Save button to save the changes.

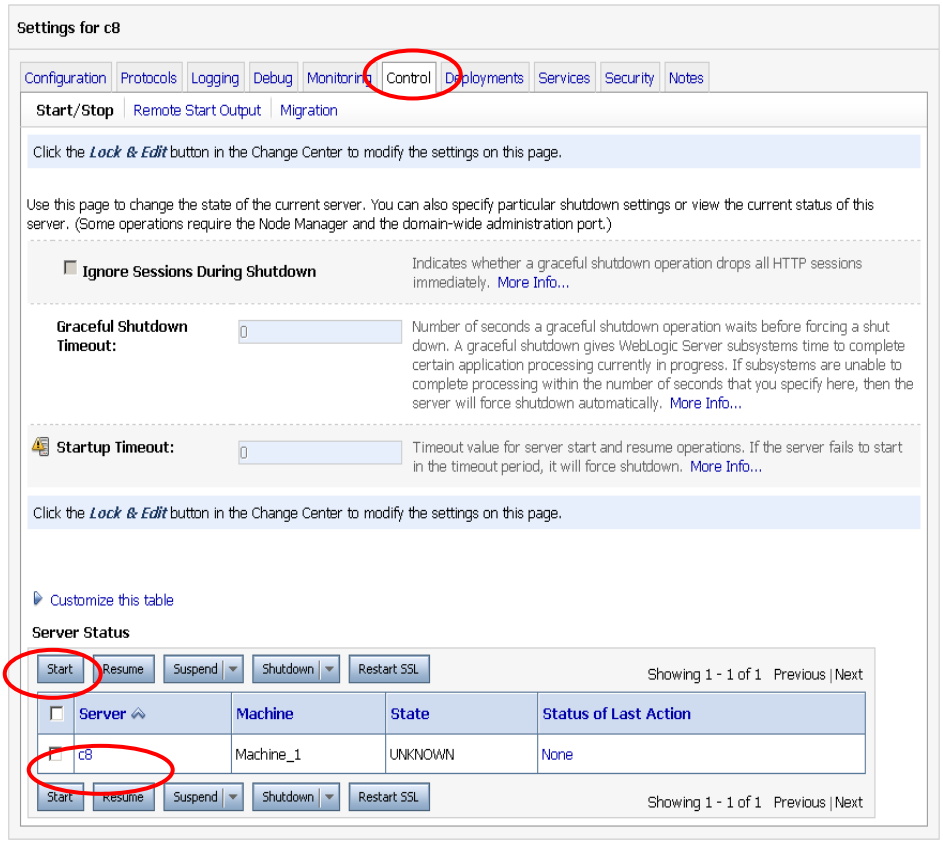


In order to activate the changes, click on the Activate Changes button. This will also release the configuration from Edit mode. Also, this is an opportunity to revert the changes. Click the Undo All changes button to restore the previous settings.

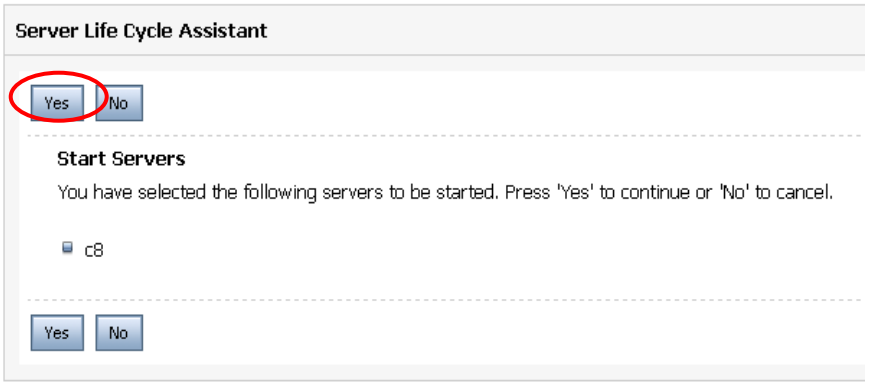


8. To start the Managed Server instance, go to the Control tab. The current state of the Managed Server is listed at the bottom of the page.

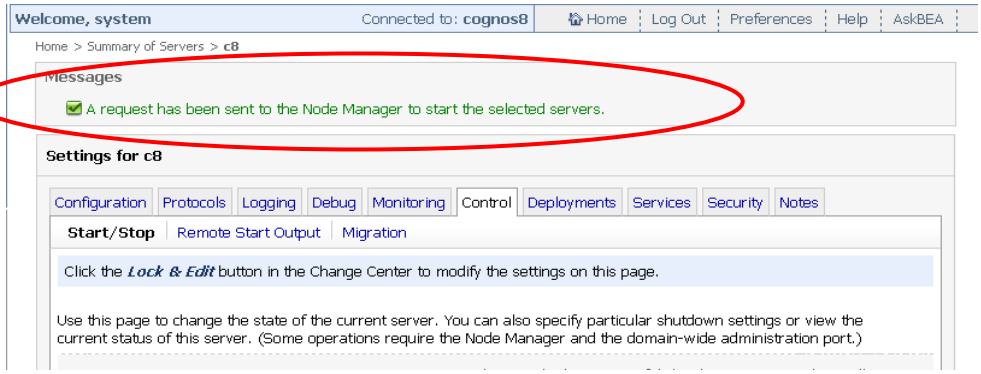
   To start the server instance, check the checkbox next to the Managed Server name and click on the Start button.



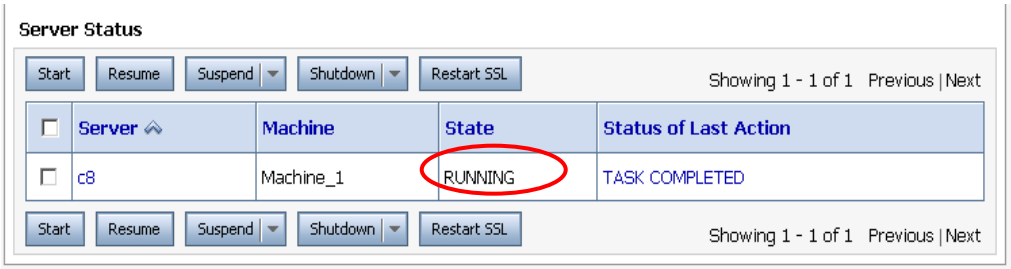
You will then be prompted to confirm the action. Click Yes to start the server instance or No to go back to the previous page.



Once the start action has been confirmed, a message acknowledging the request is displayed, saying that the request has been sent to Node Manager.



To view the status, refresh the page using the browser's refresh button.

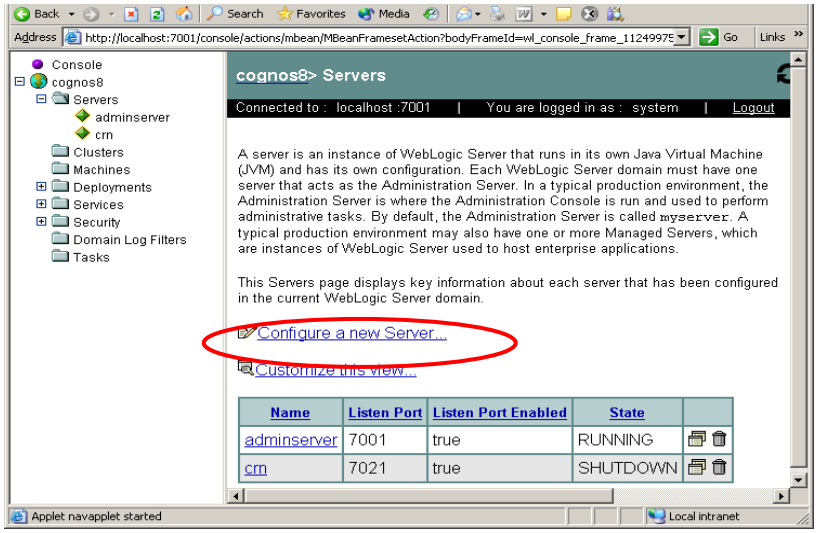


The Managed Server is now ready for use with Cognos 8.

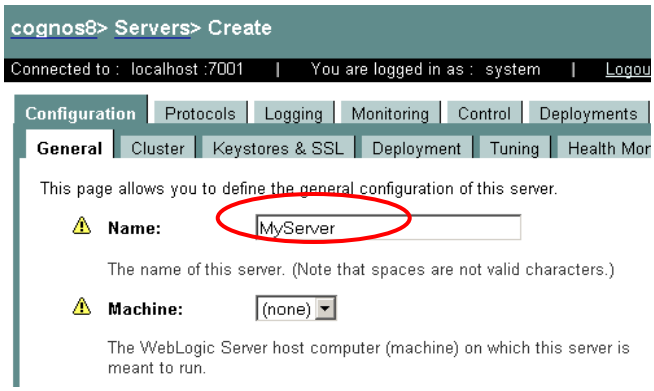
2.4 Adding Server Instances in the Administration Console

If you need to add another Managed Server instance to your domain, you can create it via the Administration Console. You do not need to create a new domain.

1. Open the Administration Console and login. Navigate to <domain_name> -> Servers. Click on the Configure a new Server… link.

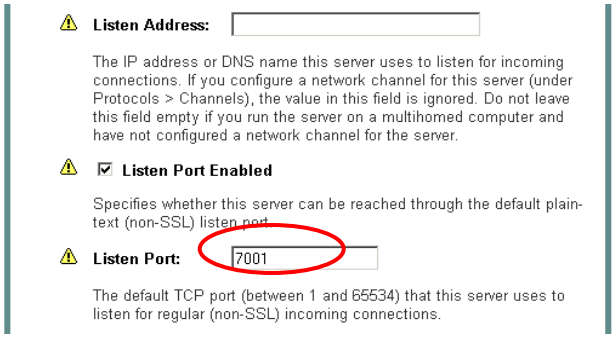


2. Replace MyServer with a unique name for the Managed Server identifier.



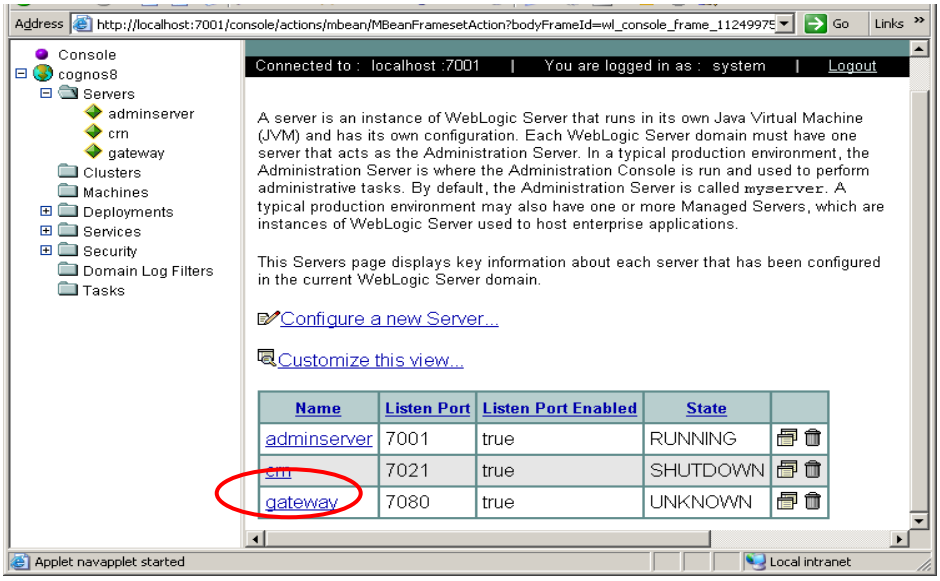
Unless you have a specific requirement, leave the Listen Address blank.

Replace 7001 with a Listen port. Use a port number that is not in use by any other application on the machine.



Scroll down to the bottom of the page and click on the Create button.

You will see the new Managed Server added to the domain configuration in the Administration Console. For example, a new Managed Server called 'gateway' with the port number of 7080 was created. Its status is listed as 'unknown', since it has not yet been activated in the domain.



3. If you intend to use the script method of starting and stopping server instances, you will need to create a new startup script for the newly created Managed Server. Follow the steps in the "Steps to Modify the WebLogic x.x.x Environment" to create a script for the new Managed Server instance and modify a copy of an existing startManagedWeblogic script.

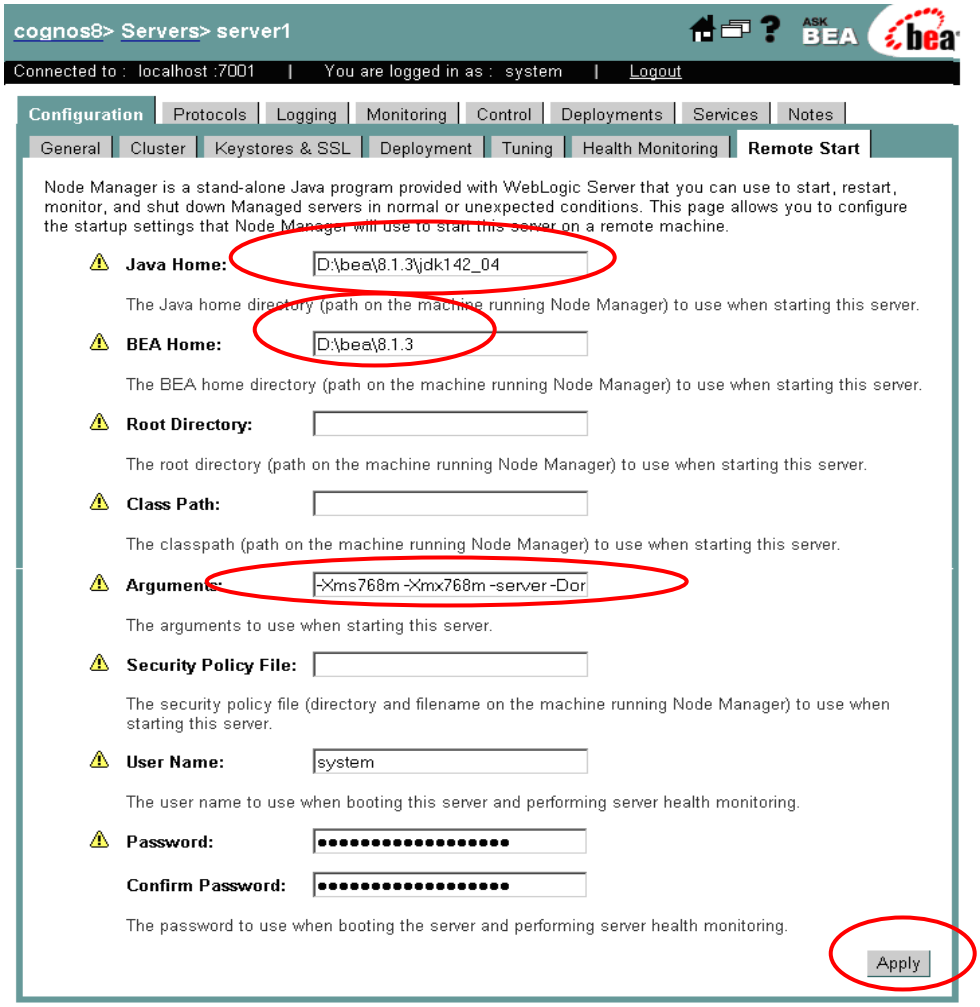
   If you intend to use the Node Manager to start and stop server instances via the Administration Console, you will need to modify the Remote Start options to set the environment settings required for running the Managed Server.

   Note: If an IBM Cognos 8 component is to be deployed to the Managed Server instance, you must modify the Arguments to include the Java parameters required for running IBM Cognos 8. If these are not set, the application will not function properly.

   The Java startup parameters include memory settings, JVM server mode setting and the org.xml.sax.driver setting. For example,

   ```
   -Xms768m -Xmx768m -server -
   Dorg.xml.sax.driver=org.apache.xerces.parsers.SAXParser -
   Dweblogic.management.discover=false
   ```

   The IBM JRE does not support the –server option. If WebLogic is running on an IBM JRE, do not add the –server option.


cognos8> Servers> server1
ASK BEA
Connected to : localhost :7001 | You are logged in as : system | Logout
Configuration
Protocols
Logging
Monitoring
Control
Deployments
Services
Notes
General
Cluster
Keystores & SSL
Deployment
Tuning
Health Monitoring
Remote Start
Node Manager is a stand-alone Java program provided with WebLogic Server that you can use to start, restart, monitor, and shut down Managed servers in normal or unexpected conditions. This page allows you to configure the startup settings that Node Manager will use to start this server on a remote machine.
Java Home:
D:\bea\8.1.3\jdk142_04
The Java home directory (path on the machine running Node Manager) to use when starting this server.
BEA Home:
D:\bea\8.1.3
The BEA home directory (path on the machine running Node Manager) to use when starting this server.
Root Directory:
The root directory (path on the machine running Node Manager) to use when starting this server.
Class Path:
The classpath (path on the machine running Node Manager) to use when starting this server.
Arguments:
-Xms768m -Xmx768m -server -Dor
The arguments to use when starting this server.
Security Policy File:
The security policy file (directory and filename on the machine running Node Manager) to use when starting this server.
User Name:
system
The user name to use when booting this server and performing server health monitoring.
Password:
Confirm Password:
The password to use when booting the server and performing server health monitoring.
Apply
View server log
View JNDI tree

3 Single Server Deployment

The following steps describe a basic single-server setup. IBM Cognos 8 has been installed with all components in one location. WebLogic Server has been installed on the same machine. Both IBM Cognos 8 and WebLogic Server need to be on the same machine in order to deploy IBM Cognos 8 as an application to WebLogic.

3.1 Pre-deployment Configuration

WebLogic Server should be started prior to deployment. You will not be able to access the Administration Console if the Administration Server isn't running. Make sure the Managed Server instance has been created and configured. It should be started prior to starting the deployment.

3.1.1 IBM Cognos 8 Setup

For detailed information regarding the configuration of IBM Cognos 8, refer to the Application Servers – Installing and Configuring the Cognos Application document and the IBM Cognos 8 user documentation.

Note:

- The JDK used when running Cognos Configuration **must** be the same as the one used for running WebLogic Server.
- The port number used by the IBM Cognos 8 Dispatcher and Content Manager is the Managed Server port that was set when the domain was created using the Domain Configuration Wizard. Do not use the Administration Server port.
- The steps outlined in ASCG05 **must** be completed prior to deploying the IBM Cognos 8 application. For example, database clients must be configured and the security provider file must be copied to the appropriate location.

3.2 Deploy IBM Cognos 8

In the Cognos Configuration tool, you can select the Expand files into a folder option when building the application files for IBM Cognos 8. The directory is the application that you will deploy to WebLogic Server. This is the recommended option for WebLogic Server. Do not use an *.ear file.

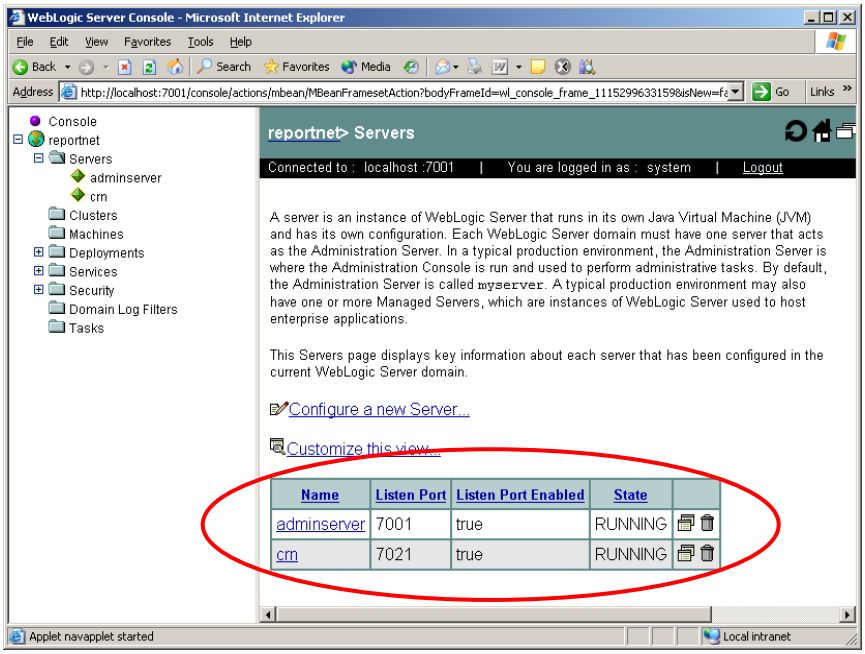
3.2.1 Steps for WebLogic 8.1.x

The following steps assume you have already expanded the application files to a folder using the Build Application Wizard in Cognos Configuration. If you did not expand the files to a folder, you should complete this step before proceeding with the deployment process.

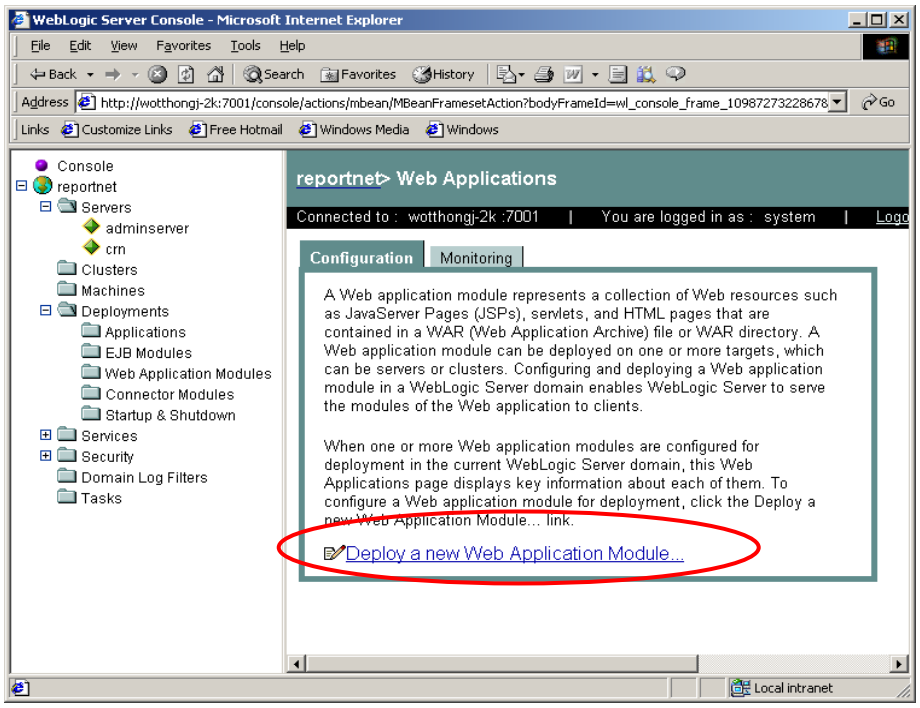
The following steps use the default application name, p2pd. The context root can be changed by using a different name when creating the war file or expanded directory. Just remember to change this in your Cognos Configuration, as well.

1. WebLogic Server should already be running. Start the WebLogic Console from the browser and login. The default address is http://localhost:7001/console. You can also use the server name. For example, http://servername:7001/console. Use the user and password created when you created the domain.

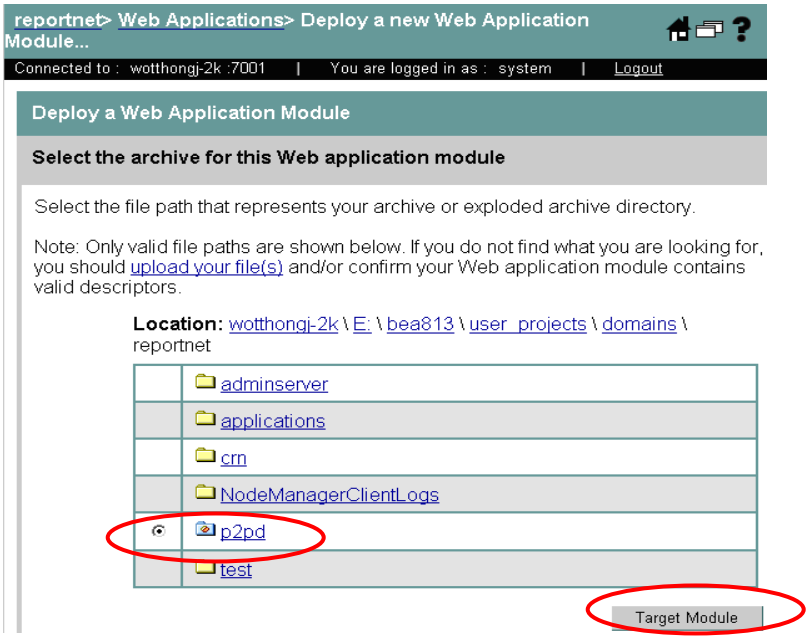
2. Make sure both your Administration Server and Managed Server are running. Select the Servers link in the left pane, and you will see the server state for each server in your WebLogic domain. They should be running.



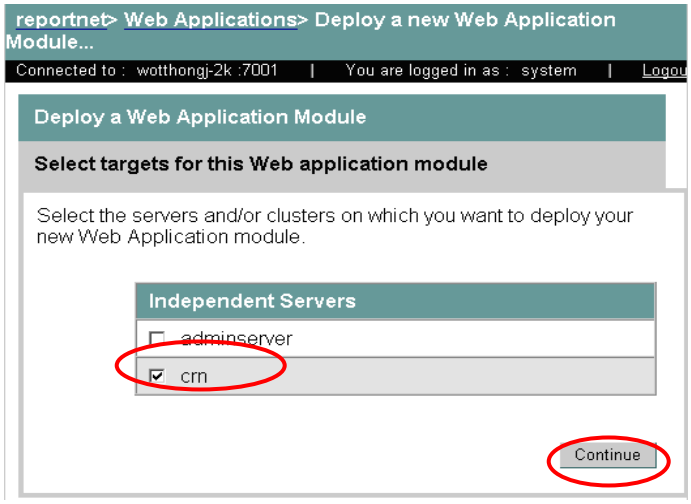
3. Deploy the application using the name of p2pd.



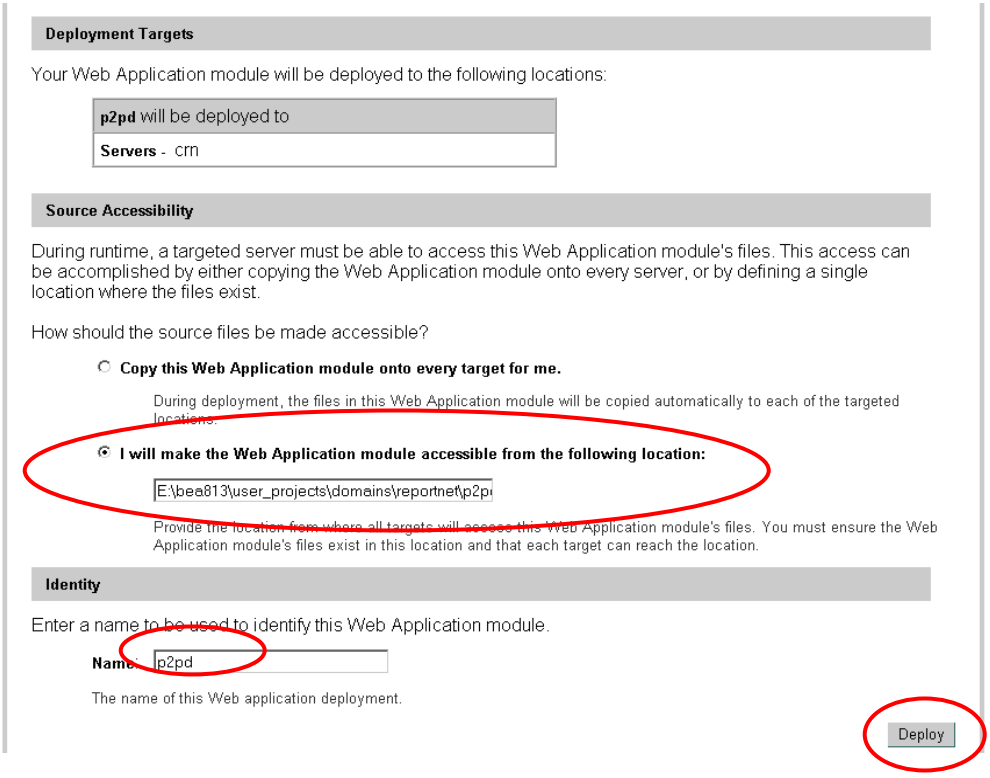
Under Deployments, select the Web Applications Modules link
Click the Deploy a new Web Application Module link



Browse to the p2pd directory. In this example, it is under %WL_HOME%\user_projects\domains, then click on Target Module button



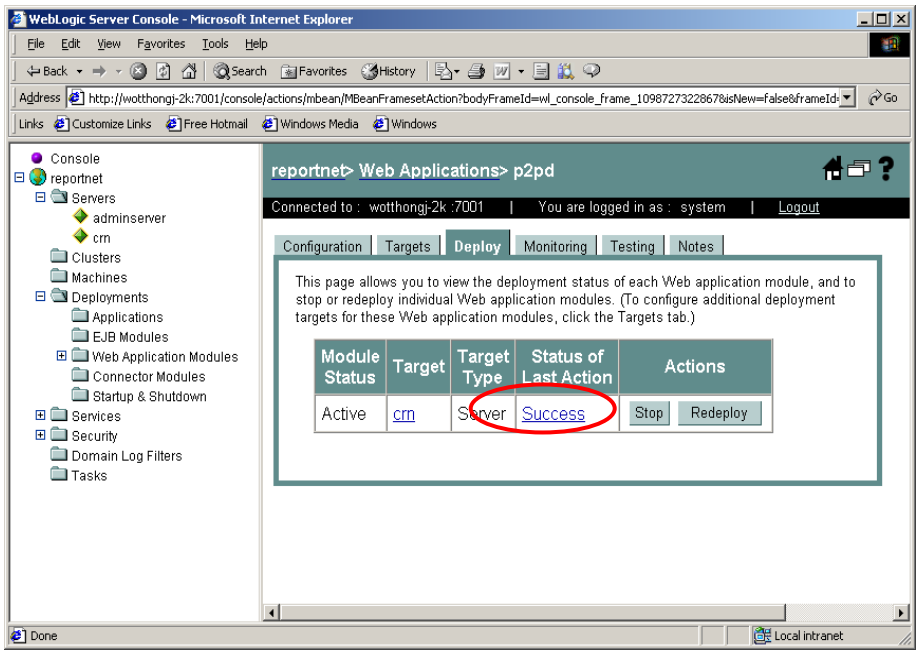
Check the target server (crn in the example above)



Select the "I will make the Web Application module accessible from the following location" option button. Enter the name if it isnot already "p2pd".

Note: If you use the default option "Copy this Web Application module onto every target for me", the application will be copied to the managed server target. If you choose to use this option, you don't need to extract the p2pd.war file into a separate location. (re. Step 2)

However, you will not be able to set the Servlet Reload Check Secs period property for performance enhancement, described later in this document.



The Deploy option should occur automatically if not click the Deploy button.

Note: this may take 5 to 10 minutes.

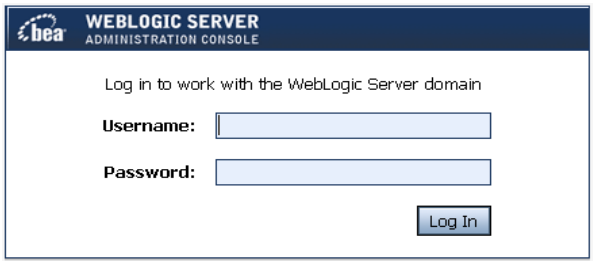
Once the deployment has completed, the status should be “Success”.

3.2.2 Steps for WebLogic 9.0.0

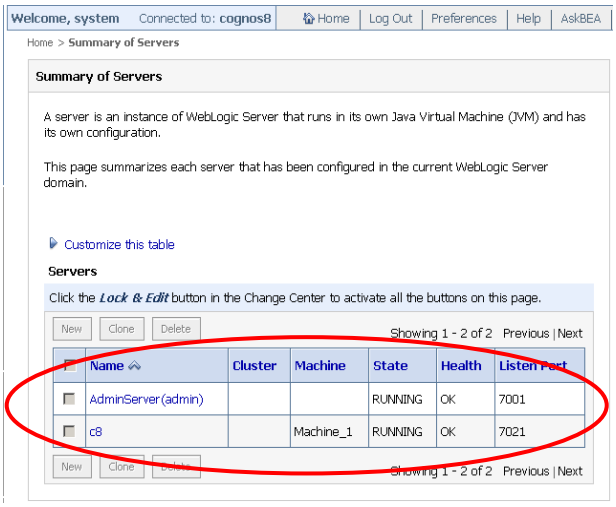
The following steps assume you have already expanded the application files to a folder using the Build Application Wizard in Cognos Configuration. If you did not expand the files to a folder, you should complete this step before proceeding with the deployment process.

The following steps use the default application name, p2pd. The context root can be changed by using a different name when creating the war file or expanded directory. Just remember to change this in your Cognos Configuration, as well.

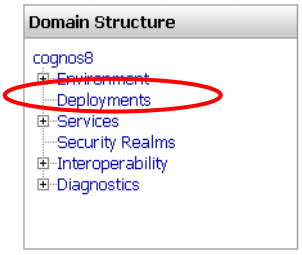
1. WebLogic Server should already be running. Start the WebLogic Console from the browser and login. The default address is http://servername:7001/console. Use the user and password created when you created the domain.



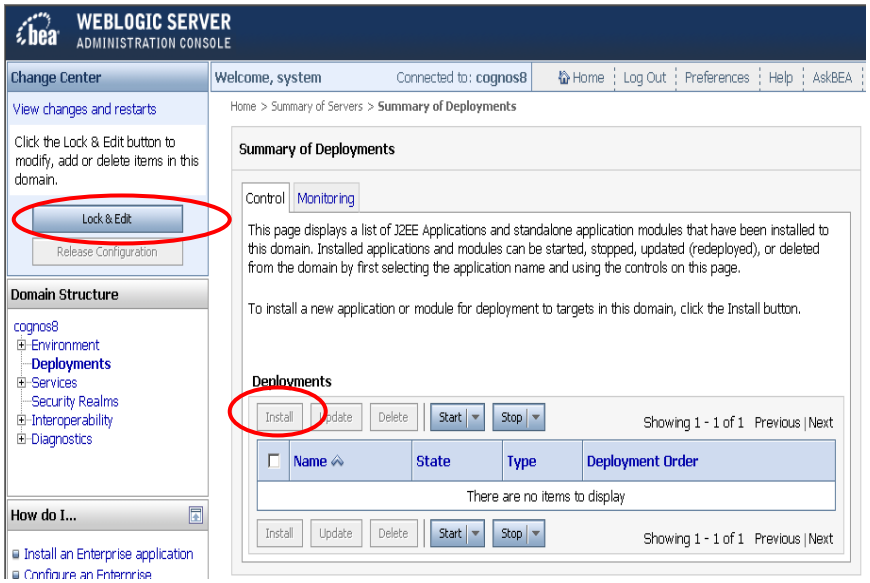
2. Make sure both your Administration Server and Managed Server are running. Select the Servers link in the left pane, and you will see the server state for each server in your WebLogic domain. They should be running.



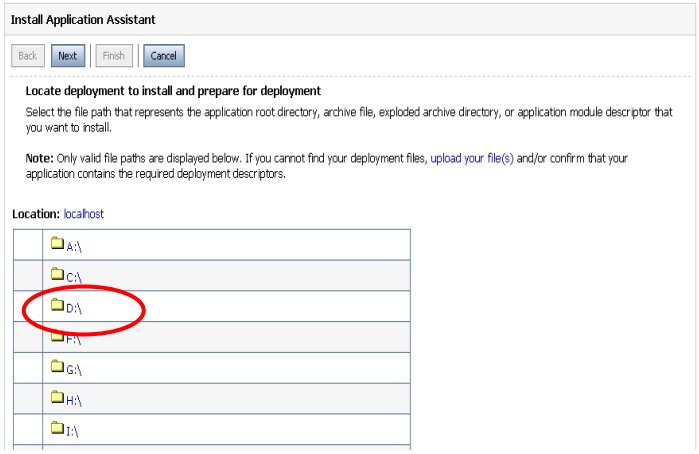
3. Deploy the application using the name of p2pd. Under Domain Structure click on Deployments.



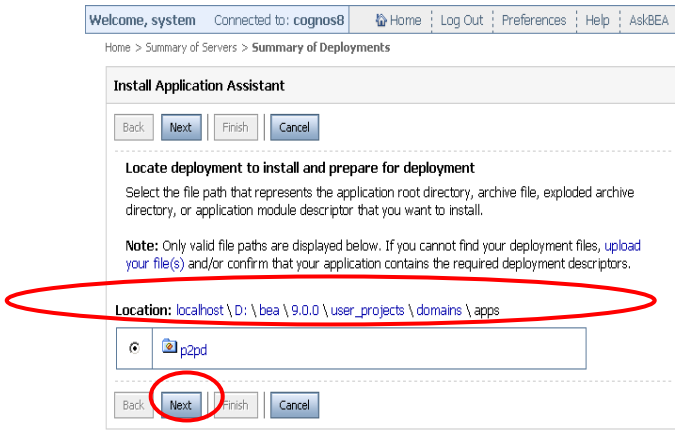
You will see the Summary of Deployments page.



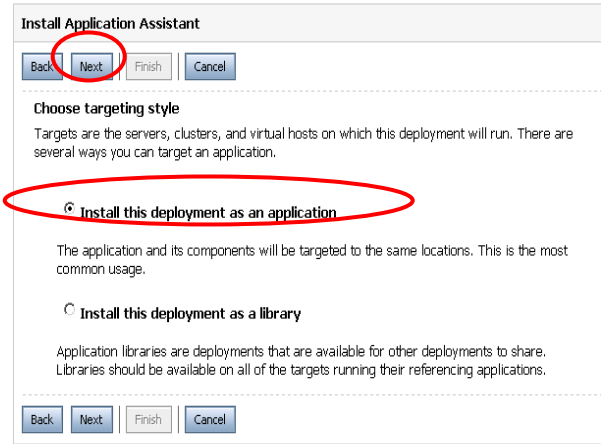
Click the Lock & Edit button to enable Edit mode.
Click the Install button to begin the deployment.



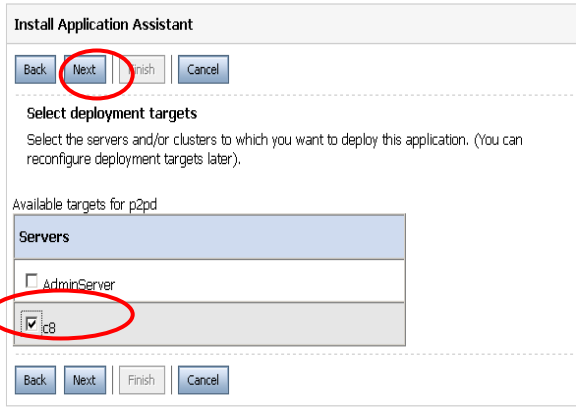
You will need to browse to the location of the p2pd application.



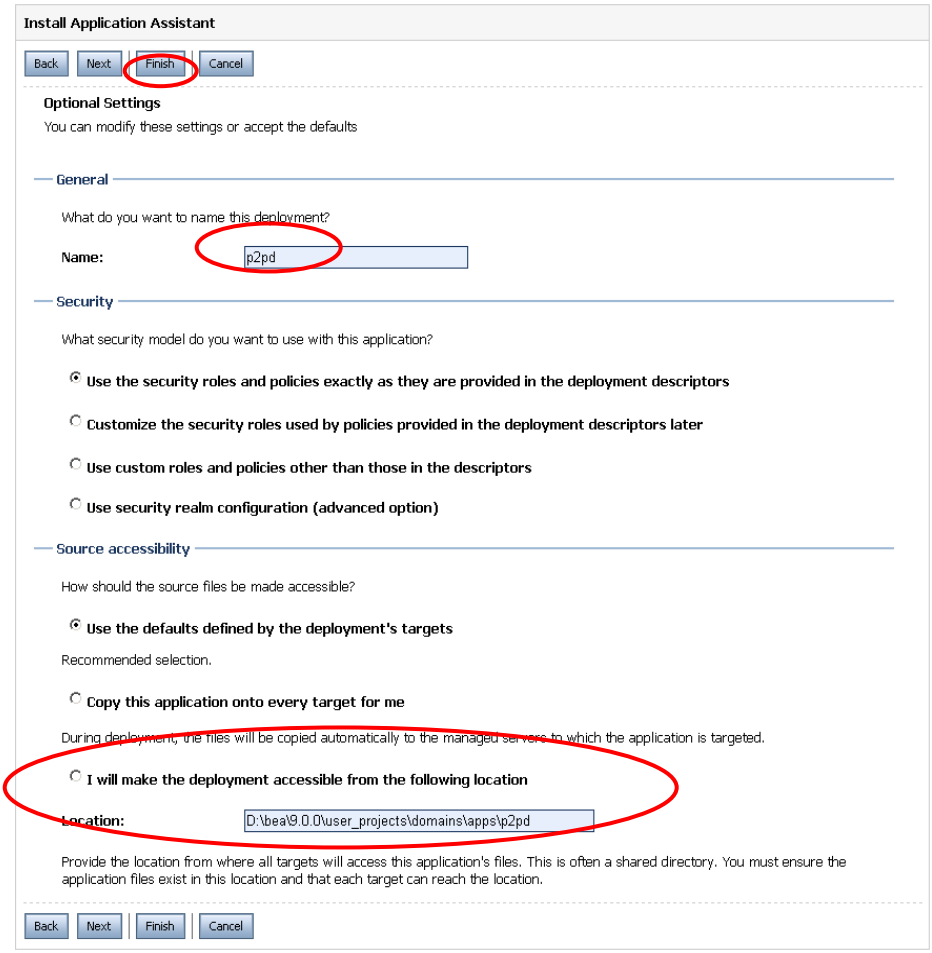
In this example, it is under %WL_HOME%\user_projects\domains\apps. Select the p2pd application and then click the Next button.



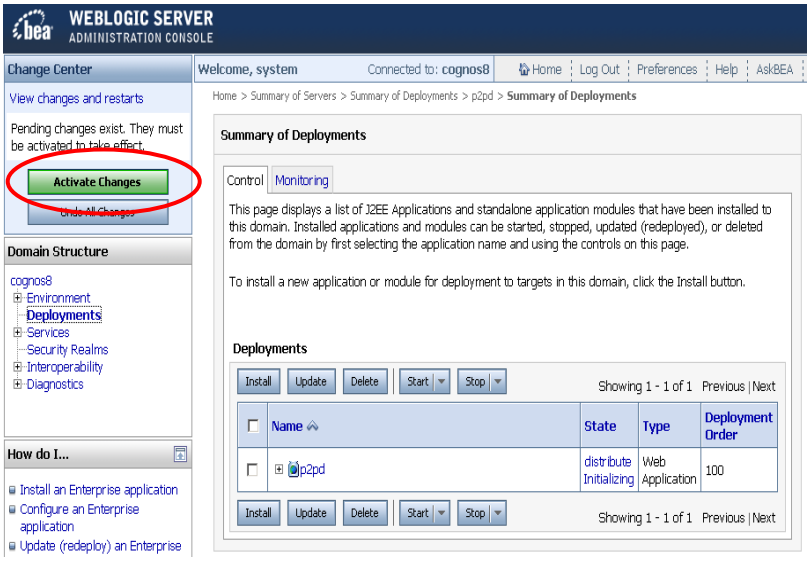
Select the Install this deployment as an application option.



Check the target server (c8 in the example above).

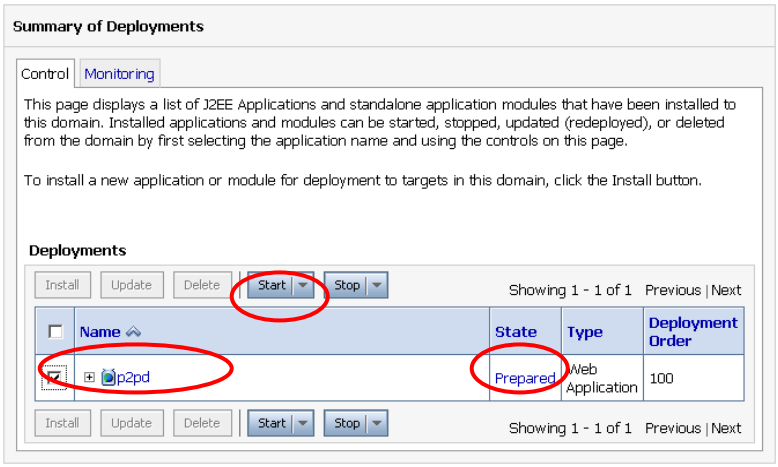


Select the “I will make the deployment accessible from the following location” option button. Enter the name, if it is not already “p2pd”.

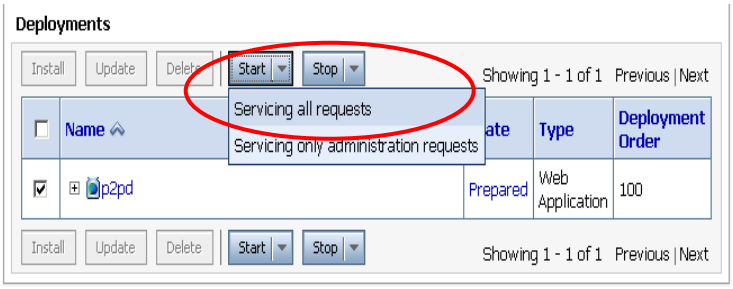


To activate the changes, click the Activate Changes button.

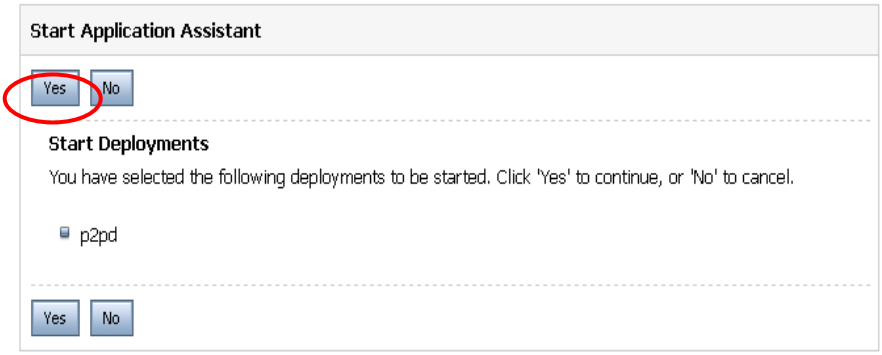
Note: this may take a few minutes.



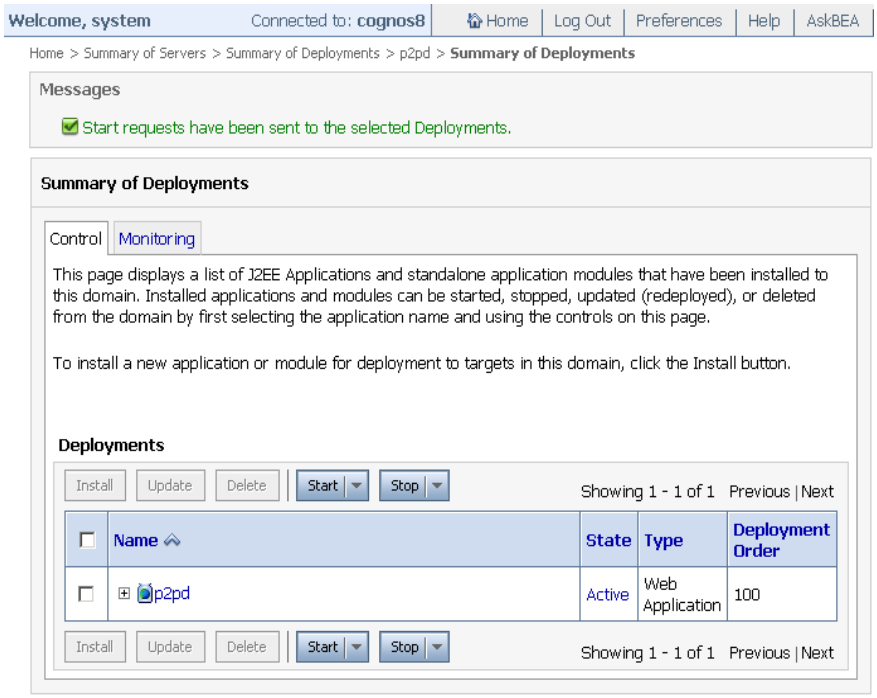
Once the deployment has completed, the status should be "Prepared". To start the application, check the p2pd checkbox. Click the Start button.



In the drop down menu, select "Servicing all requests".



You will be asked to confirm that you wish to start the application. Click the Yes button to start the application.



The request will be sent to Node Manager and the status of the application will become Active.

3.3 Post Deployment Configuration

3.3.1 Performance

There is a slight UI difference in the Administration Console between WebLogic Server 7.0.x and 8.1.x. The UI changes are more significant for version 9.x.

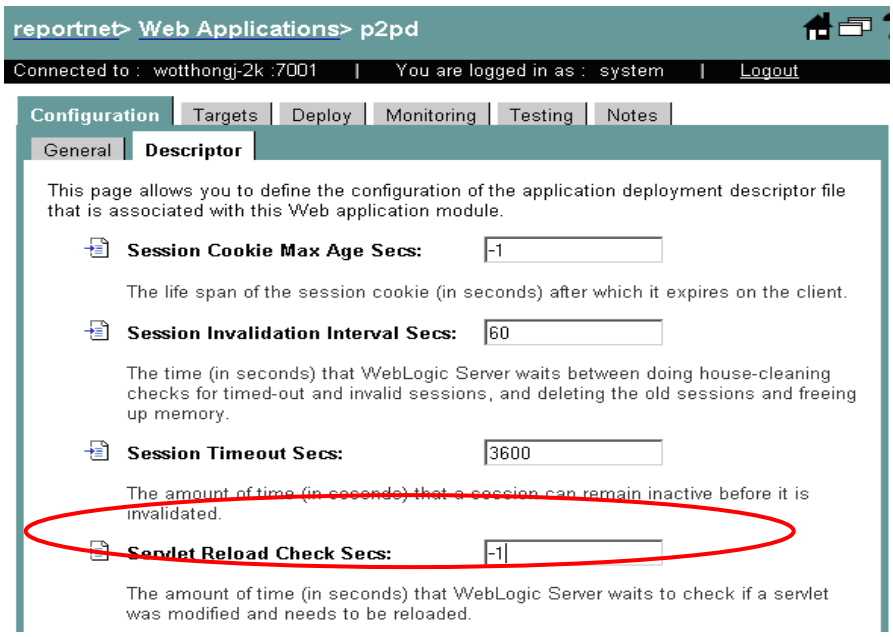
3.3.1.1 WebLogic 8.1.x

3.3.1.1.1 Servlet Reload Setting

The Servlet Reload setting defines whether a WebLogic Server will check to see if a servlet has been modified, and if it has been modified, reloads it. The -1 value tells the server never to check the servlets, 0 tells the server to always check the servlets, and the default is to check each 1 second.

A value specified in the console will always take precedence over a manually specified value. If the reload setting has not been set do the following.

Set the reload period for the Web application to -1. Apply the change.



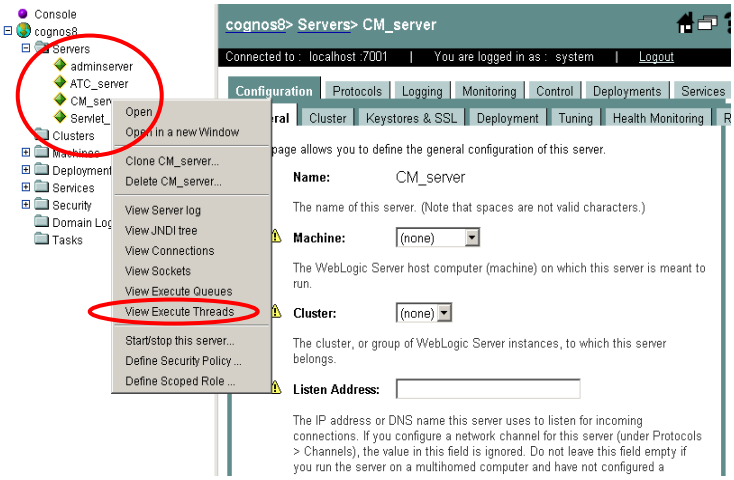
3.3.1.1.2 Modifying Default Thread Count

The default number of threads available to Execute Queues is 15 in Development mode. In Production mode the default is 25.

The thread count determines the number of simultaneous operations that can be performed by applications that use the specified execute queue. By default, all applications use the execute queue named default. You can create additional queues to exercise more control over the resources that your applications use.

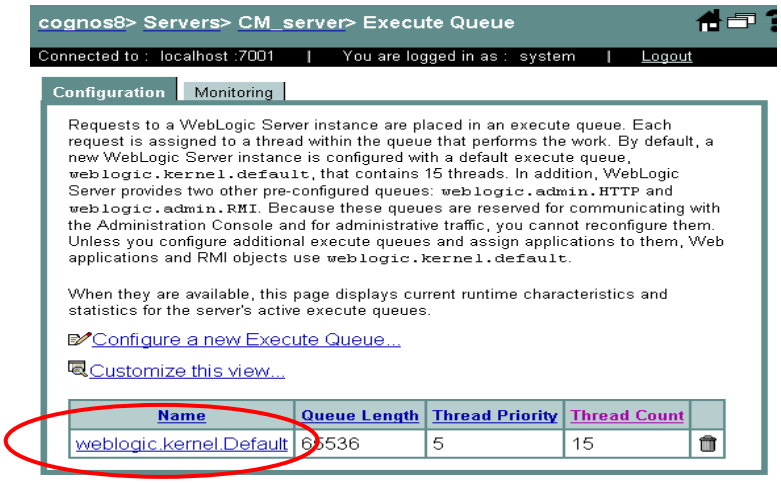
To modify the default execute queue thread count using the Administration Console:

1. Start the Administration Server if it is not already running.
2. Access the Administration Console for the domain.
3. Expand the Servers node in the left pane to display the servers configured in your domain. Right-click the name of the server instance that contains the execute queue you want to configure, and then select View Execute Queues on the pop-up menu to display a table of execute queues that can be modified.



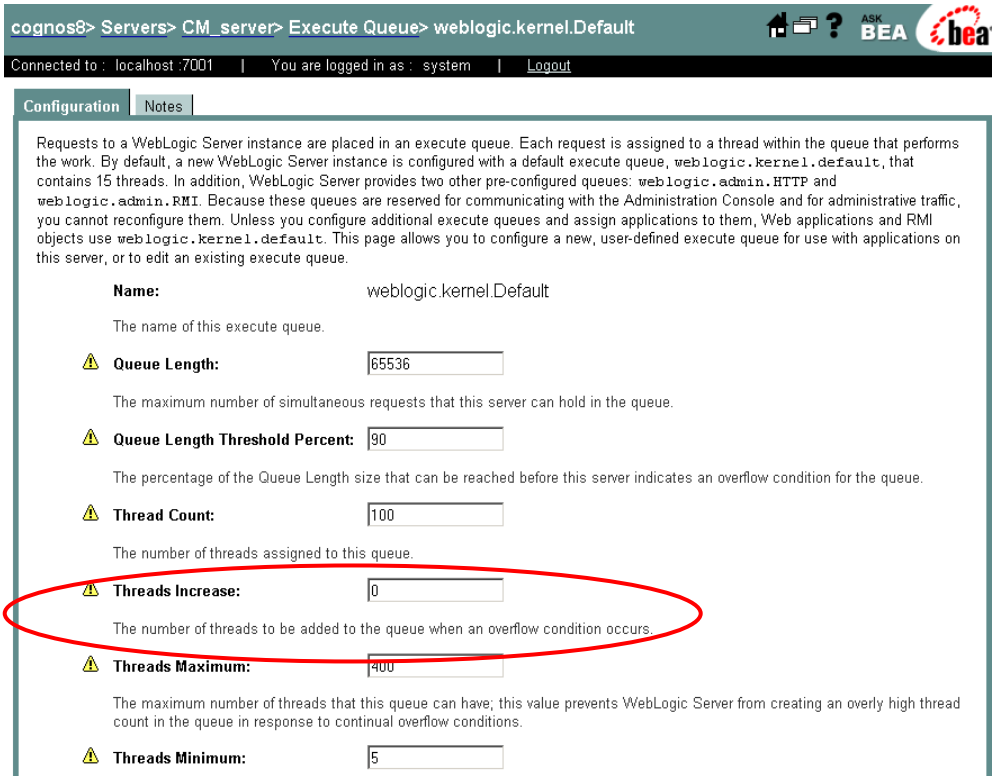
Note: You can only modify the default execute queue for the server or a user-defined execute queue.

4. In the Name column, click directly on the default execute queue name to display the Configuration tab for modifying execute queues.



5. On the Configuration Tab, locate the Thread Count value and increase or decrease it, as appropriate. The appropriate value depends on the resources available on your machine. You are limited by the power of your processor and the available memory which is consumed by each thread.

Note: if the yellow icon is flashing the server must be stopped and restarted.



6. Click Apply to save your changes. Restart the server to enable the new settings.

   For more information regarding modifying the thread count and other performance tuning options, consult the WebLogic documentation. The following link may be useful.

   http://e-docs.bea.com/wls/docs81/perform/WLSTuning.html#1139296

3.3.1.2 WebLogic 9.0.0

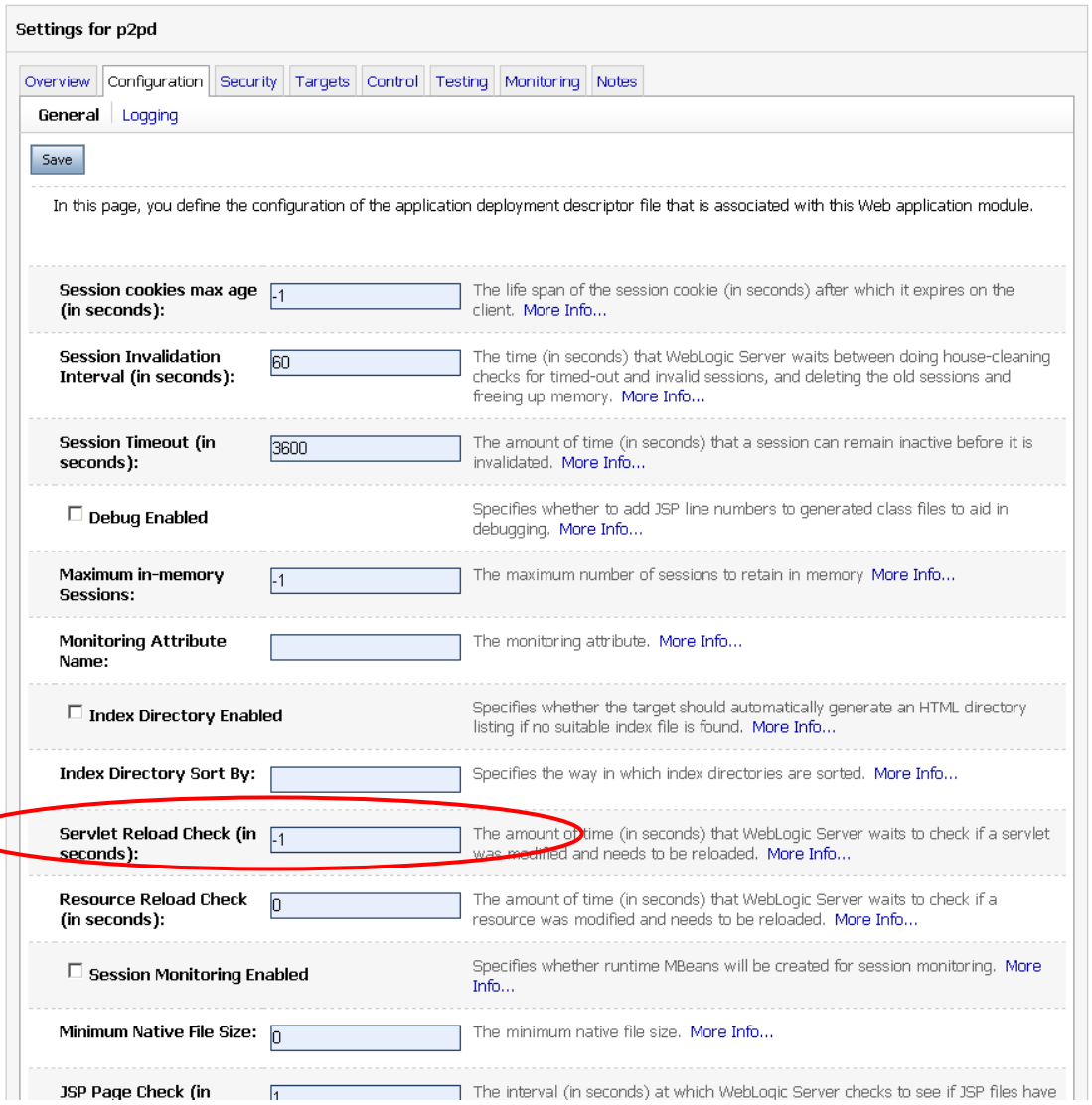
3.3.1.2.1 Servlet Reload Setting

The Servlet Reload setting defines the amount of time in seconds WebLogic Server waits to check if a servlet has been modified, and if it has been modified, reloads it. The -1 value tells the server never to check the servlets, 0 tells the server to always check the servlets, and the default is to check each 1 second.

A value specified in the console will always take precedence over a manually specified value. If the reload setting has not been set do the following.

1. Login to the Administration Console and stop the Managed Server instance hosting the p2pd application.
2. Navigate to the Summary of Deployments and select the p2pd application.
3. In the Setttings for p2pd view, select the Configuration tab. Click the Lock & Edit button to enter Edit mode.
4. Set the Servlet reload check period to -1. Save the change.

5. The Save Deployment Plan Assistant appears. Select p2pd and click Finish. Click the Activate Changes button.



3.3.1.2.2 Modifying Default Thread Count

In WebLogic 9.0, Work Managers are introduced to help manage workload on Managed Servers. Work is prioritized and threads are allocated based on an execution model that takes into account administrator-defined parameters and actual run-time performance and throughput.

WebLogic Server now uses a single thread pool, in which all types of work are executed. Work is prioritized based on rules defined by the administrator, and run-time metrics, including the actual time it takes to execute a request and the rate at which requests are entering and leaving the pool.

The common thread pool changes its size automatically to maximize throughput. The queue monitors throughput over time and based on history, determines whether to adjust the thread count. For example, if historical throughput statistics indicate that a higher thread count increased throughput, WebLogic

increases the thread count. Similarly, if statistics indicate that fewer threads did not reduce throughput, WebLogic decreases the thread count. This new strategy implemented by BEA is meant to make it easier for administrators to allocate processing resources and manage performance, avoiding the effort and complexity involved in configuring, monitoring, and tuning custom executes queues.

To manage work in your applications, you define one or more of the following Work Manager components:

- Fair Share Request Class:
- Response Time Request Class:
- Min Threads Constraint:
- Max Threads Constraint:
- Capacity Constraint
- Context Request Class

There are three types of Work Managers, each one characterized by its scope and how it is defined and used. They are:

- The default Work Manager
- Global Work Managers
- Application-scoped Work Managers

To handle thread management and perform self-tuning, WebLogic Server implements a default Work Manager. This Work Manager is used by an application when no other Work Managers are specified in the application's deployment descriptors.

In many situations, the default Work Manager may be sufficient for most application requirements. WebLogic Server's thread-handling algorithms assign each application its own fair share by default. Applications are given equal priority for threads and are prevented from monopolizing them.

You can override the behaviour of the default Work Manager by creating and configuring a global Work Manager called default. This allows you to control the default thread-handling behaviour of WebLogic Server.

Refer to the link below for more information regarding Work Managers.

http://e-docs.bea.com/wls/docs91/config_wls/self_tuned.html

3.3.1.2.2.1 Enabling Execute Queues in WebLogic 9

WebLogic Server 8.1 used Execute Queues to handle thread management, allowing you to create thread-pools to determine how workload was handled. WebLogic Server 9 still provides Execute Queues for backward compatibility, primarily to facilitate application migration. However, new application development should utilize Work Managers to perform thread management more efficiently.

You can enable Execute Queues in the following ways:

- Using the command line option

```
-Dweblogic.Use81StyleExecuteQueues=true
```

- Setting the Use81StyleExecuteQueues property via the Kernel MBean in config.xml.

Enabling Execute Queues disables all Work Manager configuration and thread self tuning. Execute Queues behave exactly as they did in WebLogic Server 8.1. Refer to the link below for more information regarding user-defined execute queues.

http://e-docs.bea.com/wls/docs91/perform/appb_queues.html#1042349

When enabled, Work Managers are converted to Execute Queues based on the following rules:

- If the Work Manager implements a minimum or maximum threads constraint, then an Execute Queue is created with the same name as the Work Manager. The thread count of the Execute Queue is based on the value defined in the constraint.
- If the Work Manager does not implement any constraints, the the global default Execute Queue is used.

When an application is migrated from WebLogic Server 8.1, any Execute Queues defined in the server configuration before migration will still be present. WebLogic Server does not automatically convert the Execute Queues to Work Managers.

When an 8.1 application implementing Execute Queues is deployed on WebLogic Server 9.x, the Execute Queues are created and used handle thread management for requests. However, only those requests whose dispatch-policy maps to an Execute Queue will take advantage of this feature.

To modify the default execute queue thread count using the Administration Console:

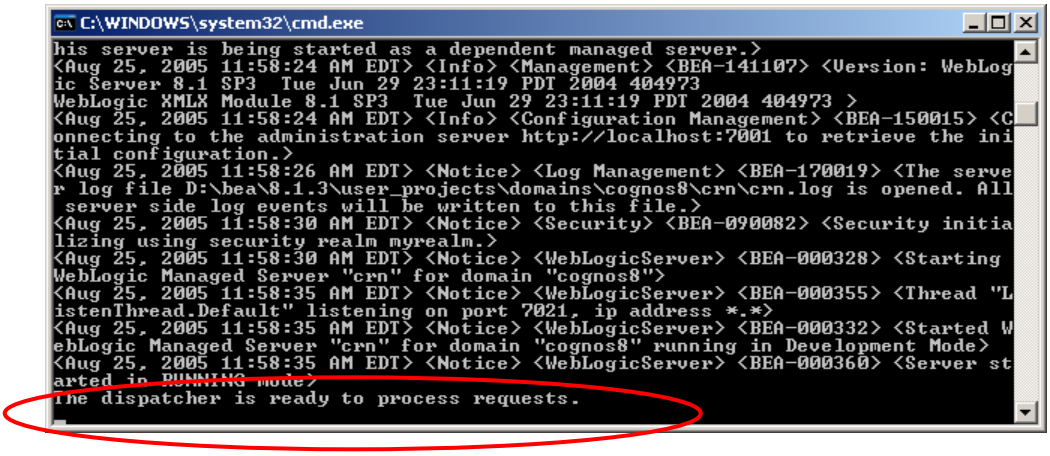
1. Start the Administration Server if it is not already running and login to the Administration Console for the domain.
2. If you have not already done so, in the Change Center of the Administration Console, click Lock & Edit.
3. In the left pane of the console, expand Environment > Servers.
4. On the Summary of Servers page, select the server instance for which you will configure thread detection behavior.
5. On the Configuration > Queues tab, select the execute queue for which you will modify the default thread count.

   Note: You can only modify the default execute queue for the server or a user-defined execute queue.

5. Locate the Thread Count value and increase or decrease it, as appropriate.
6. Click Save.
7. To activate these changes, in the Change Center of the Administration Console, click Activate Changes. Not all changes take effect immediately—some require a restart.
8. You must reboot the server to use the new thread detection behavior values.

3.4 Verify the Deployed Application

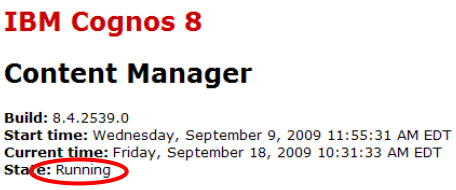
Once the deployment has completed successfully, you should see a message stating “The dispatcher is ready to process requests” in the Managed Server’s console, if started via the startup script. This indicates that CM has started and the Dispatcher for Cognos 8 has been registered to CM and has started. You should also see an active BIBusTKServerMain process running when consulting the active processes on the machine. Use the appropriate utility for your OS to verify this.



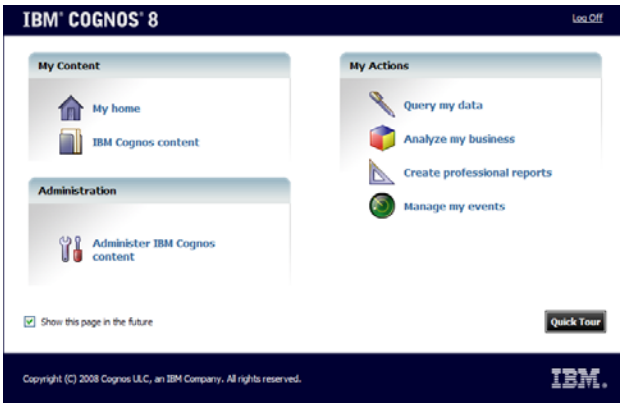
Note: The server console output to the screen is only available when the server instances are started using the scripts. They are not available if the server instances are started via the Administration Console using Node Manager.

To verify that Content Manager has started, you can check the cogserver.log or open a browser and type in the Content Manager URI. The default is http://localhost:<port_number>/p2pd/servlet if you used p2pd as the context root.

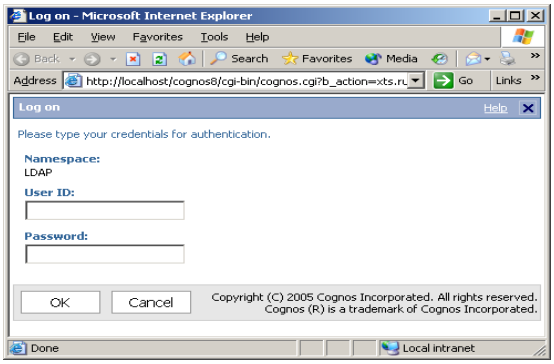
For example, the Managed Server to which IBM Cognos 8 has been deployed is running on port number 7021. To verify that CM is running, you would enter http://localhost:7021/p2pd/servlet. If the Content Store was successfully created and CM started, then you should see a page similar to the figure below.



One last test to verify that the deployment was successful is to access the Portal. Open a new browser session and enter the URL for IBM Cognos Connection. The default is http://localhost/cognos8, if you are using a web server. IBM Cognos Connection should open and you should see the Welcome page.



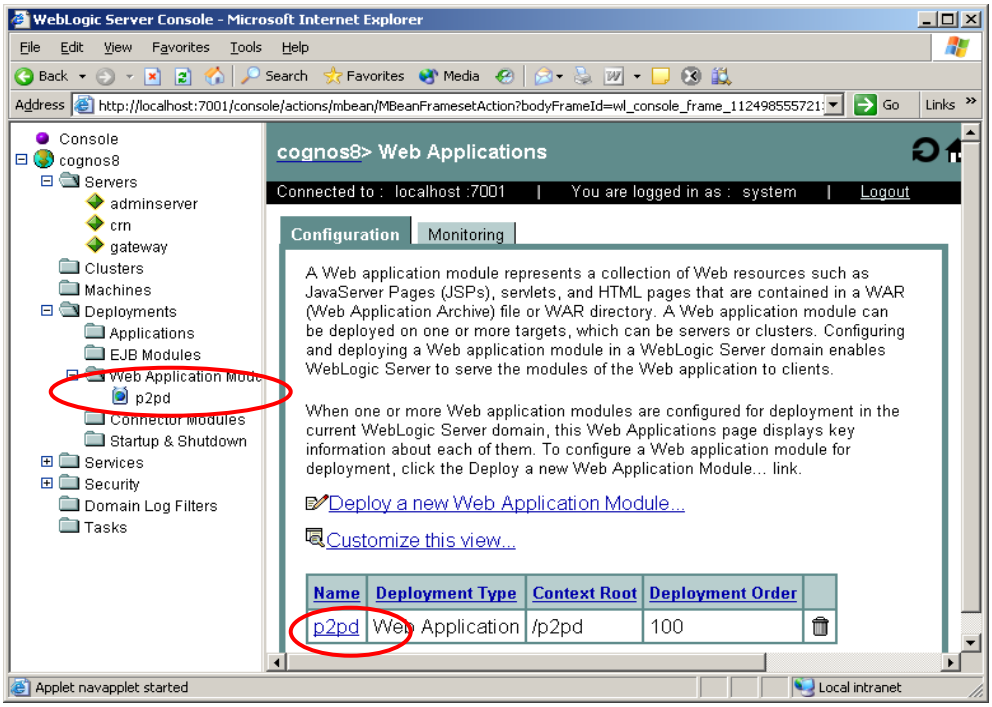
If a 3rd party namespace (eg. LDAP) was configured in Cognos Configuration, then you would be prompted with the login page.



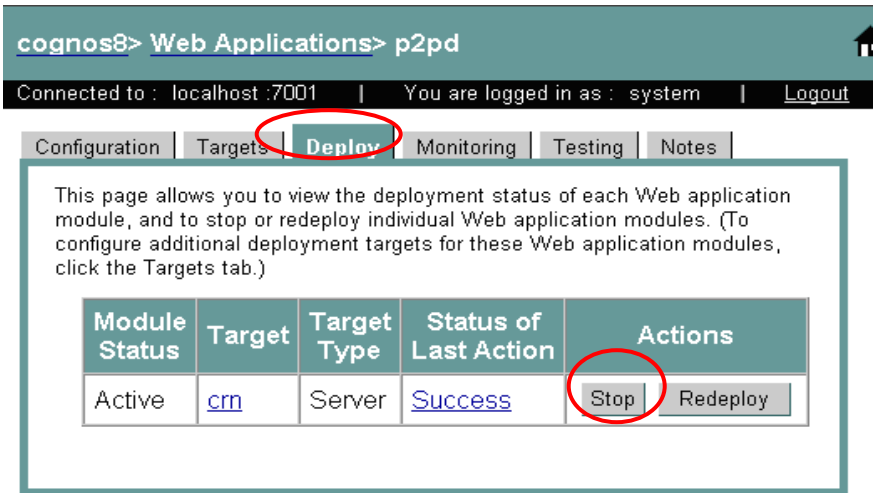
3.5 Undeploy IBM Cognos 8

Complete the following steps to remove IBM Cognos 8 from your WebLogic Server installation.

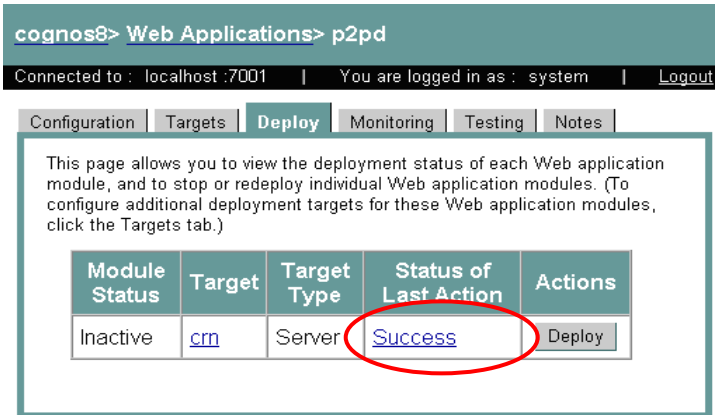
1. While the WebLogic Server processes are running, open the Administration Console and login.
2. In the left pane of the Administrator Console, under the domain, expand Deployments and then highlight Web Applications. In the right pane, click on the p2pd application.



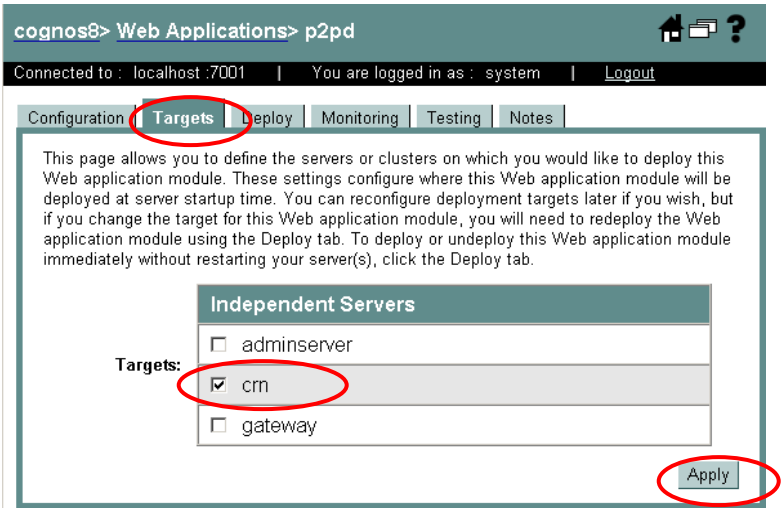
3. Click on the Deploy tab. For WebLogic 8.1.x, click on the Stop button.



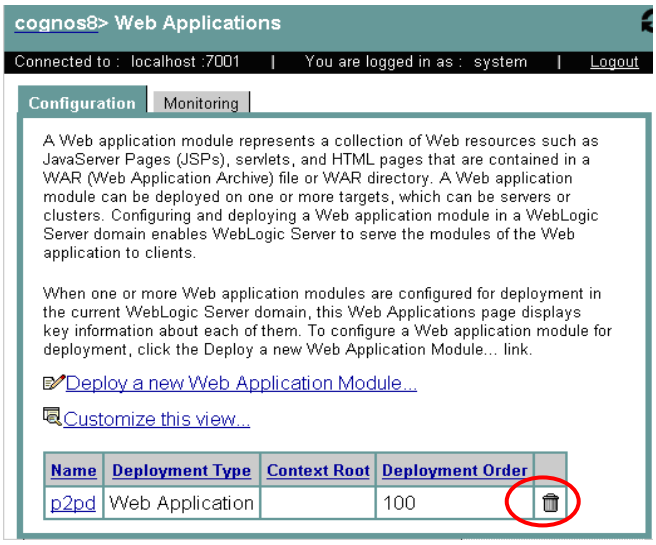
Wait for the action to complete.



4. Once the application has been rendered inactive, navigate to the Targets tab. Remove the Managed Server as a target by unchecking the checkbox next to the Managed Server.



5. In the right pane, click on <domain name> -> Deployments -> Web Applications. Click on the trash can for the p2pd application to remove it from the configuration.

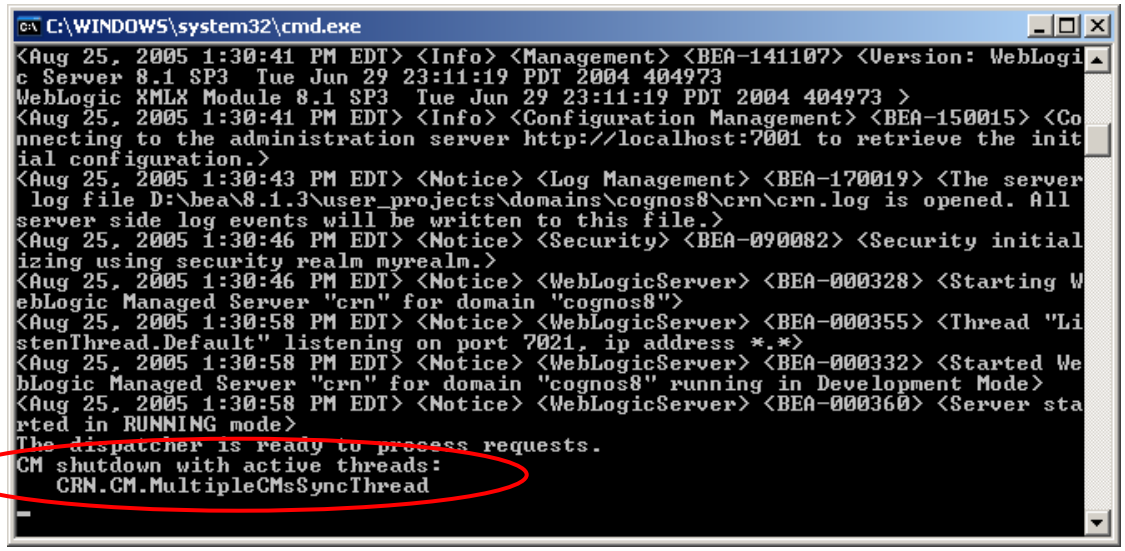


You will be prompted to confirm that you wish to remove the application, select Yes and continue.

You will see a message written in the Managed Server console that indicates that CM is shutting down with active threads.

```
CM shutdown with active threads:
   CRN.CM.MultipleCMsSyncThread
```

This is normal and no action is required, other than shutting down the Managed Server instance.



Note: You can remove the application from the configuration without completing the initial cleanup steps, but an error message will be written to the Administration Server. The errors are not serious in nature. You should restart the WebLogic processes to release memory and threads.

4 Distributed Server Deployment

The following steps will describe setting up each IBM Cognos 8 component on separate Managed Server instances.

If deploying each IBM Cognos 8 component to a separate server, you will need to have WebLogic Server installed on each machine where you wish to run a IBM Cognos 8 component. You should make note of the Managed Server name and port number to be used on each machine.

If deploying each component on the same WebLogic Server install, make sure you have created Managed Server instances for each component and make note of the names and port numbers. You cannot deploy each component to the same Managed Server instance, since classes can only be loaded once.

Content Manager (CM) must be up and running before configuring the Application Tier Components (ATC) or Gateway (GW) distributed installs using IBM Cognos Configuration. The context root for CM and ATC must be the same. In this example, all the necessary steps to start the CM instance will be completed prior to performing any ATC or GW steps.

For this example, we will install each IBM Cognos 8 component in a separate location on the machine.

- Content Manager is installed in D:\cognos\cm
- Application Tier Components is installed in D:\cognos\atc
- Gateway is installed in D:\cognos\gw

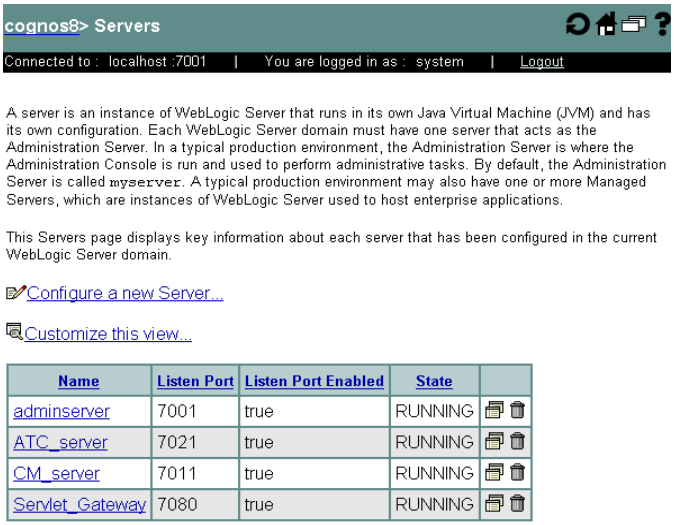
Once installed, do not rename or change the installation directories for the IBM Cognos 8 components.

The WebLogic domain for this example is cognos8 and has the following configuration.

- Administration Server on port 7001
- CM_server on port 7011
- ATC_server on port 7021
- Servlet_Gateway on port 7080

Each Cognos component must be deployed to a separate Managed Server instance for distributed scenarios. You cannot deploy two IBM Cognos applications to the same Managed Server instance.

The same JDK must be used for configuring IBM Cognos 8 components and running the WebLogic Server instances. You cannot mix versions.



The steps to undeploy the IBM Cognos 8 components are the same as those used for the single server deployment. The only difference is that each component must be undeployed from its Managed Server instance.

4.1 Content Manager

You should have Content Manager (CM) installed in a separate location. The CM component is the first component to be configured and deployed. You cannot continue onto the other IBM Cognos 8 components until CM is up and running.

The deployment steps followed in the Administration Console are the same as those used for a single server deployment.

4.1.1 Pre-deployment Configuration

WebLogic Server should be started prior to starting the deployment. You will not be able to access the Administration Console if the Administration Server isn't running. Make sure the Managed Server instance has been created and configured.

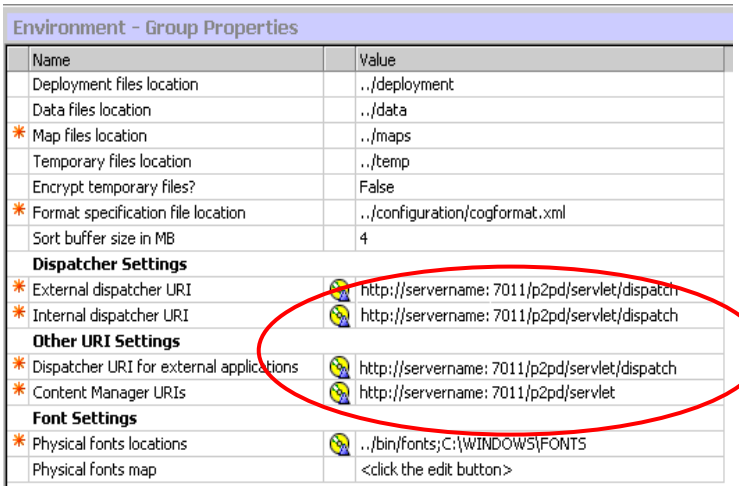
Each IBM Cognos component must be deployed to a separate Managed Server instance for distributed scenarios.

4.1.1.1 Cognos 8 Setup

For detailed information regarding the configuration of IBM Cognos 8, refer to the Application Servers – Installing and Configuring the IBM Cognos Application document and the IBM Cognos 8 user documentation.

The steps to configure and deploy CM are similar to those for a full single-server install.

- The same environment variables need to be set.
- The JDK used when running Cognos Configuration **must** be the same as the one used for running WebLogic Server.
- The steps outlined in ASCG05 **must** be completed prior to deploying the IBM Cognos 8 application. For example, database clients must be configured and the security provider file must be copied to the appropriate location.
- The port number used by the IBM Cognos 8 Content Manager is the Managed Server port that was set when the domain was created using the Domain Configuration Wizard or when the new Server was added to the domain configuration via the Administration Console. Do not use the Administration Server port.



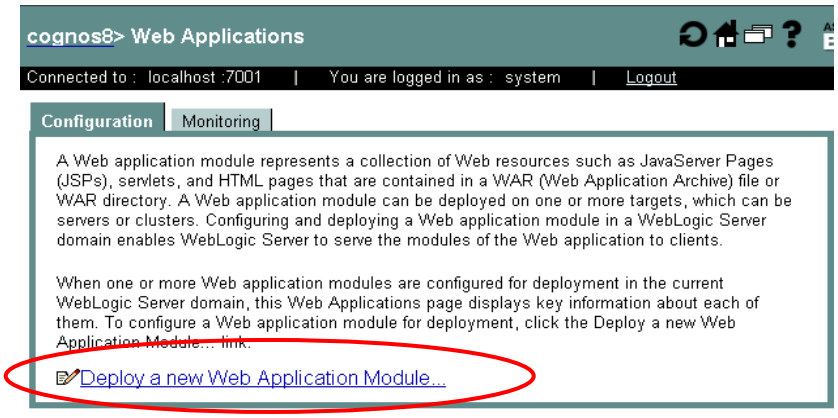
- The application was created using the Build Application Wizard. You should use the Expand files into a folder option.
- The database instance to be used for the IBM Cognos 8 Content Store must be created and available. The user and password must also be created with suitable privileges.

4.1.2 Deployment

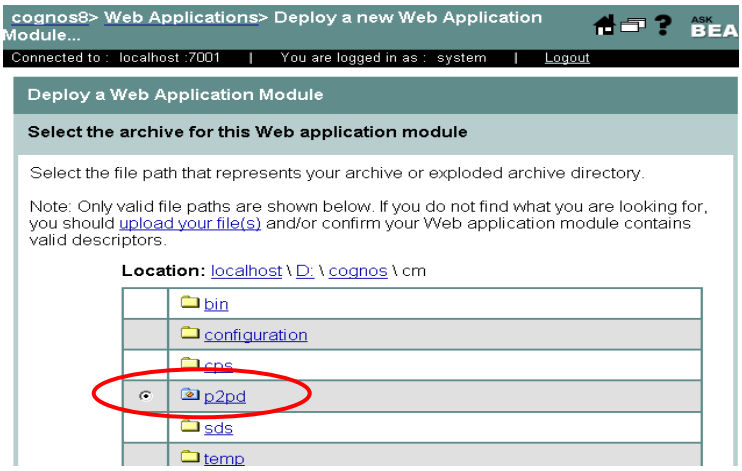
The deployment sequence in the Administration Console is the same as the steps for a single server setup. The difference here is that a deployment will occur for each component. Only the WebLogic 8.1.x dialogs will be displayed for this example. Refer to the single server steps for the WebLogic 7.0.x dialogs.

The default context root, p2pd, will be used for this example. It is important that you use the same context root for the CM and ATC component applications. The Servlet Gateway can use a different context root.

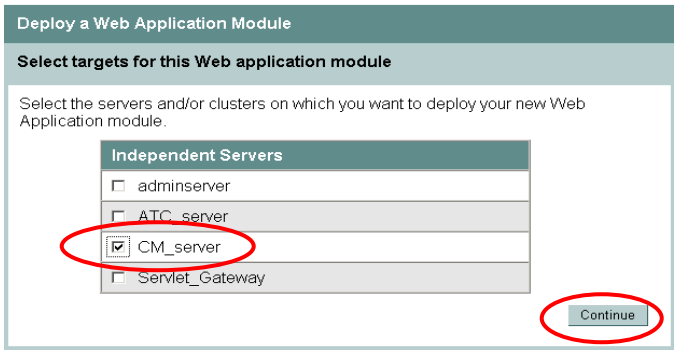
1. WebLogic Server should already be running. Start the WebLogic Console from the browser and login. The default address is http://localhost:7001/console. You can also use the server name. For example, http://servername:7001/console. User the user and password created when you created the domain.
2. Under Deployments, select the Web Applications Modules link. Click the Deploy a new Web Application Module link.



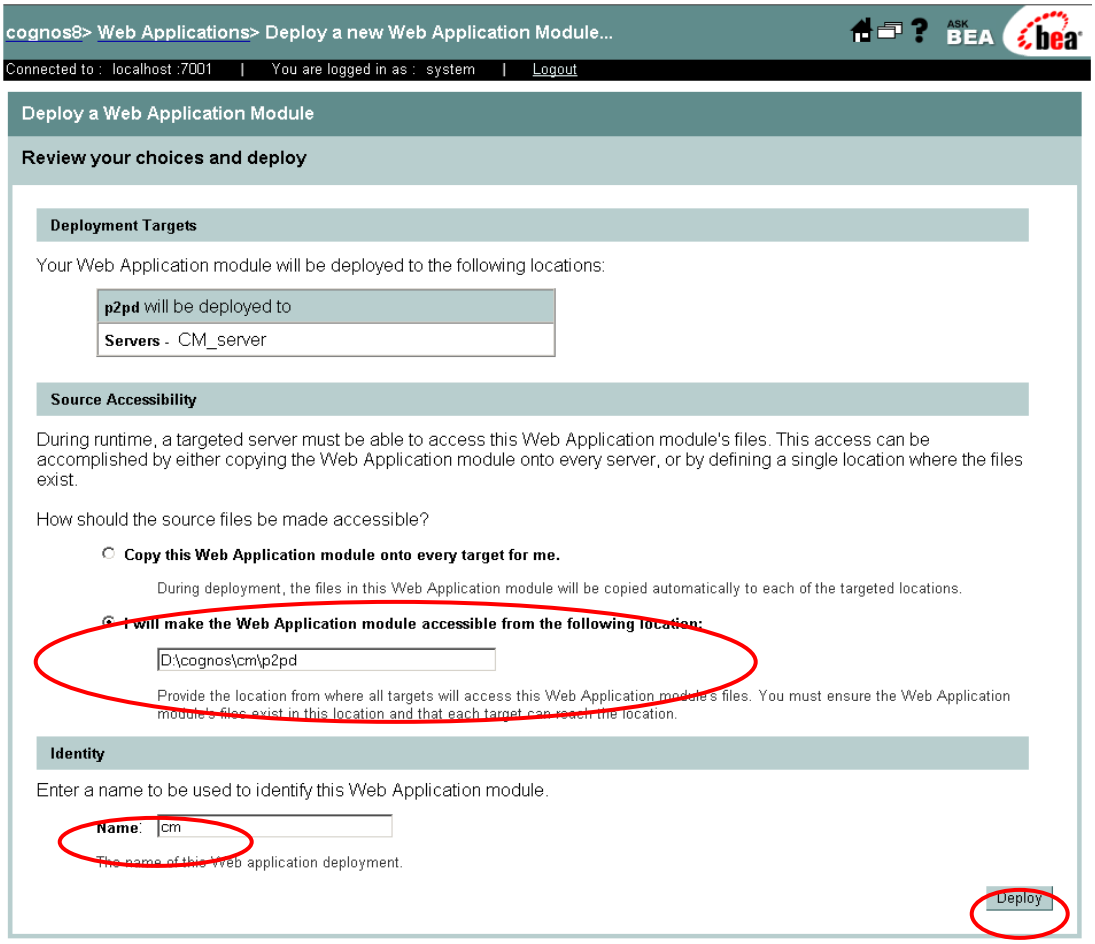
3. Browse to the p2pd application and select it. In this example, it is under D:\cognos\cm. Click on Target Module button.



4. Check the target server. In this example, the target server is CM_server. Click on Continue

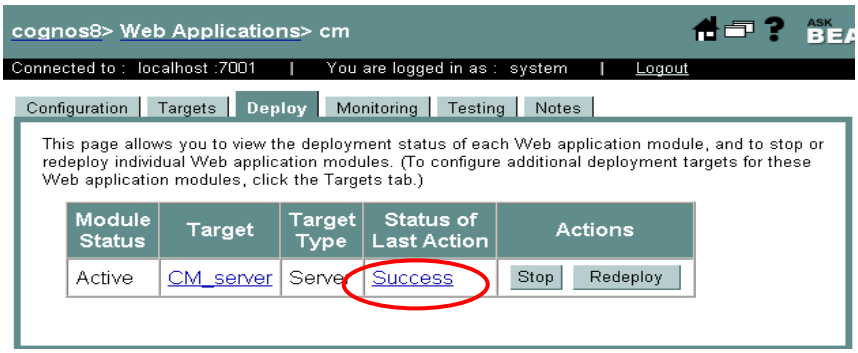


5. Select the "I will make the Web Application module accessible from the following location" option button. Under the Identify section, the default name of the application is p2pd. Change this to cm. Each application must be identified with a unique name. This does not change the context root of the application. Click the Deploy button to continue.



Note: If you use the default option "Copy this Web Application module onto every target for me", the application will be copied to the managed server target. If you choose to use this option, you don't need to extract the p2pd.war file into a separate location. However, you will not be able to set the Servlet Reload Check Secs period property for performance enhancement.

6. The Deploy option should occur automatically if not click the Deploy button. This may take a few minutes. Once the deployment has completed, the status should be "Success".

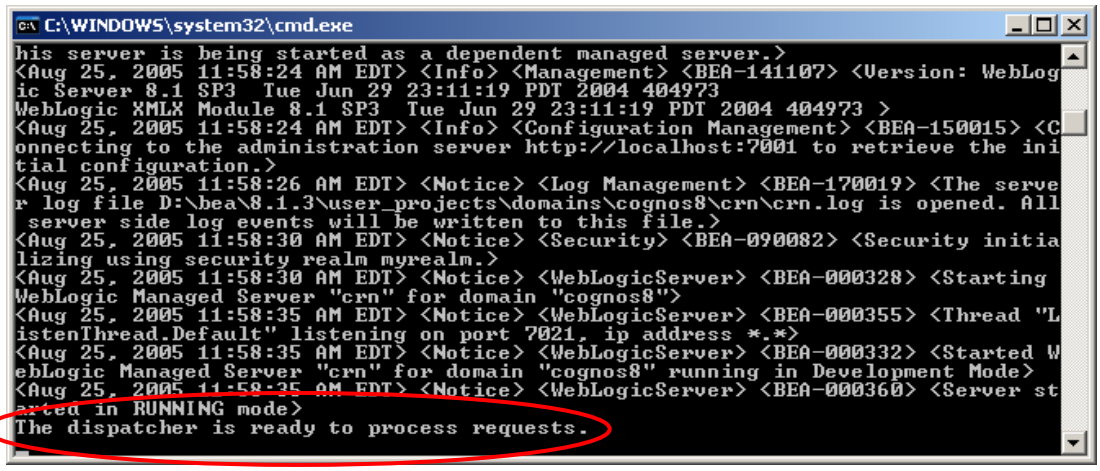


4.2 Post Deployment Configuration

The post deployment steps are the same as those used for the single server installation.

4.2.1 Verify the Deployed Application

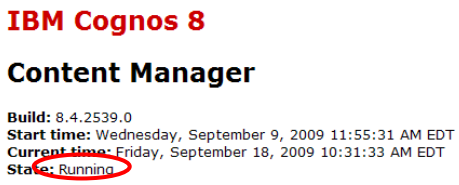
Once the deployment has completed successfully, you should see a message stating "The dispatcher is ready to process requests" in the Managed Server's console. This indicates that CM has started and the Dispatcher for IBM Cognos 8 has been registered to CM and has started. You should also see an active BIBusTKServerMain process running when consulting the active processes on the machine. Use the appropriate utility for your OS to verify this.



Note: The server console output to the screen is only available when the server instances are started using the scripts. They are not available if the server instances are started via the Administration Console using Node Manager.

To verify that Content Manager has started, you can check the cogserver.log or open a browser and type in the Content Manager URI. The default is http://localhost:<port_number>/p2pd/servlet if you used p2pd as the context root.

For example, the Managed Server to which Content Manager has been deployed is running on port number 7021. To verify that CM is running, you would enter http://localhost:7021/p2pd/servlet. If the Content Store was successfully created and CM started, then you should see a page similar to the figure below.



If CM is not running, you must rectify the problem before you can continue with the setup of the other IBM Cognos 8 components.

4.3 Application Tier Components

You should have the Application Tier Components (ATC) installed in a separate location. The ATC component is the second component to be configured and deployed before the Gateway component can be configured and deployed. You cannot configure ATC until CM is up and running. You cannot configure GW until ATC has been configured and deployed.

4.3.1 Pre-deployment Configuration

WebLogic Server should be started prior to starting the deployment. You will not be able to access the Administration Console if the Administration Server isn't running. Make sure the Managed Server instance has been created and configured.

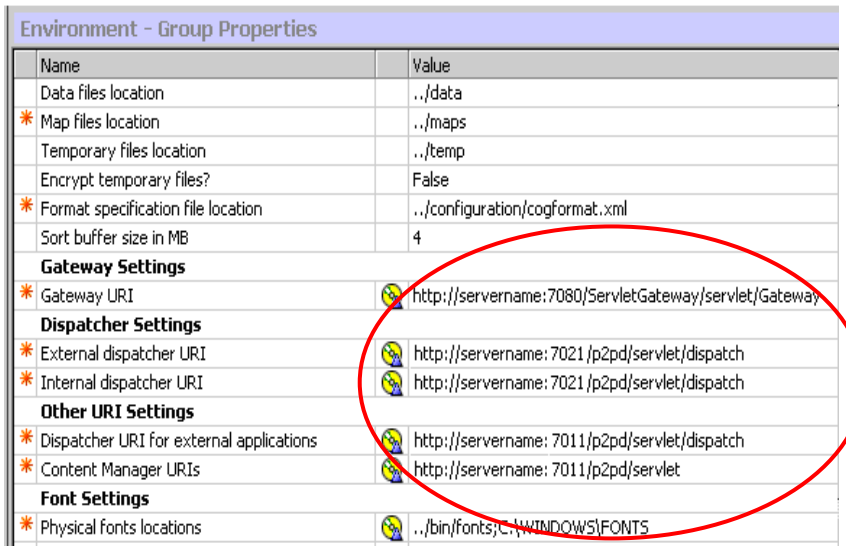
Each Cognos component must be deployed to a separate Managed Server instance for distributed scenarios.

4.3.1.1 IBM Cognos 8 Setup

For detailed information regarding the configuration of IBM Cognos 8, refer to the Application Servers – Installing and Configuring the IBM Cognos Application document and the IBM Cognos 8 user documentation.

The steps to configure and deploy ATC are similar to those for a full single-server install.

- The same environment variables need to be set.
- The JDK used when running IBM Cognos Configuration **must** be the same as the one used for running WebLogic Server.
- The steps outlined in ASCG05 **must** be completed prior to deploying the IBM Cognos 8 application. For example, database clients must be configured and the security provider file must be copied to the appropriate location.
- The port number used by the IBM Cognos 8 Dispatcher is the Managed Server port that was set when the domain was created using the Domain Configuration Wizard or when the new Server was added to the domain configuration via the Administration Console. Do not use the Administration Server port.



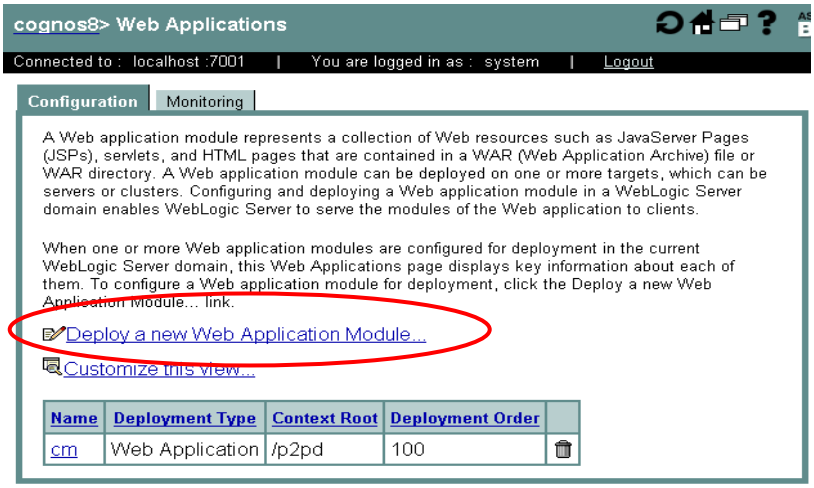
- Make sure CM is running, before configuring ATC using the Cognos configuration tool. CM must also be running before starting the deployment in WebLogic Server.
- The application was created using the Build Application Wizard. You should use the Expand files into a folder option.

4.3.2 Deployment

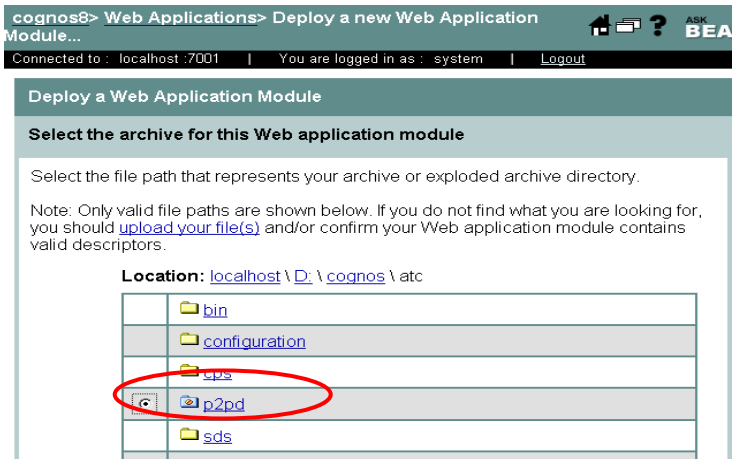
The deployment sequence in the Administration Console is the same as the steps for a single server setup. The difference here is that a deployment will occur for each component. Only the WebLogic 8.1.x dialogs will be displayed for this example.

The default context root, p2pd, will be used for this example. It is important that you use the same context root for the CM and ATC component applications. The Servlet Gateway can use a different context root.

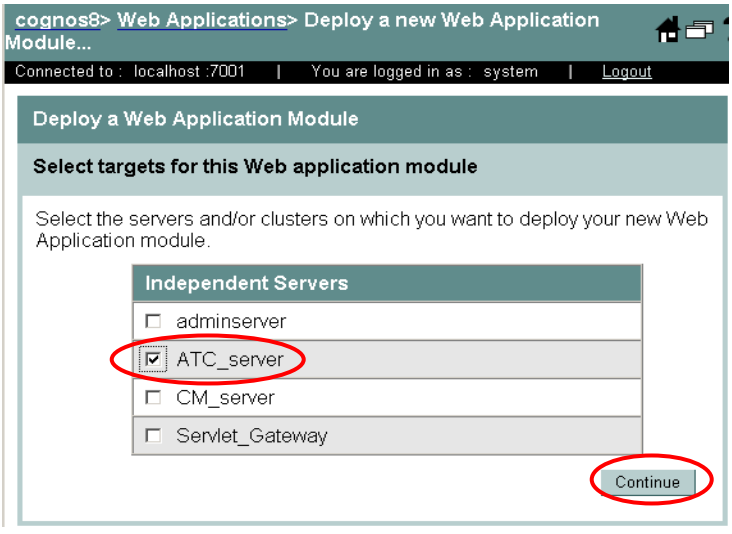
1. WebLogic Server should already be running. Start the WebLogic Console from the browser and login. The default address is http://localhost:7001/console. You can also use the server name. For example, http://servername:7001/console. User the user and password created when you created the domain.
2. Under Deployments, select the Web Applications Modules link. Click the Deploy a new Web Application Module link. You should already have the CM application deployed and running.



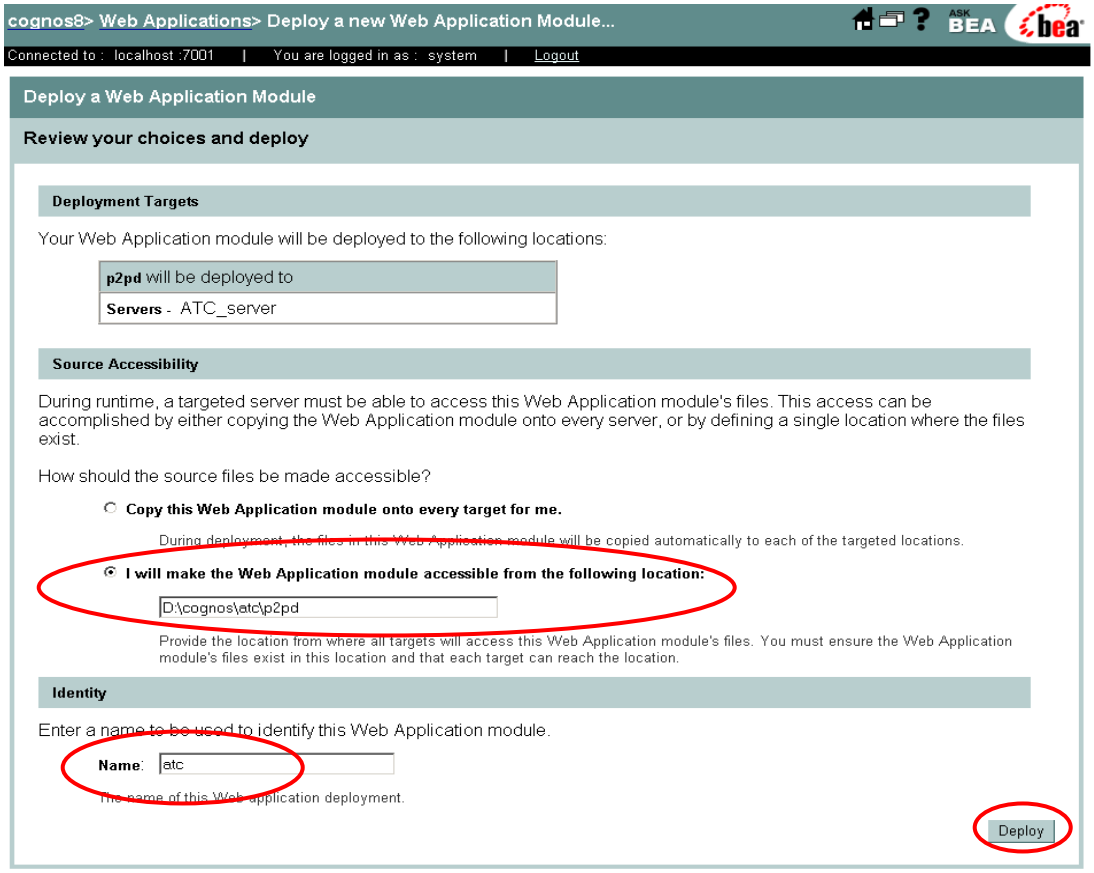
3. Browse to the p2pd application and select it. In this example, it is under D:\cognos\atc. Click on Target Module button.



4. Check the target server. In this example, the target server is ATC_server. Click on Continue

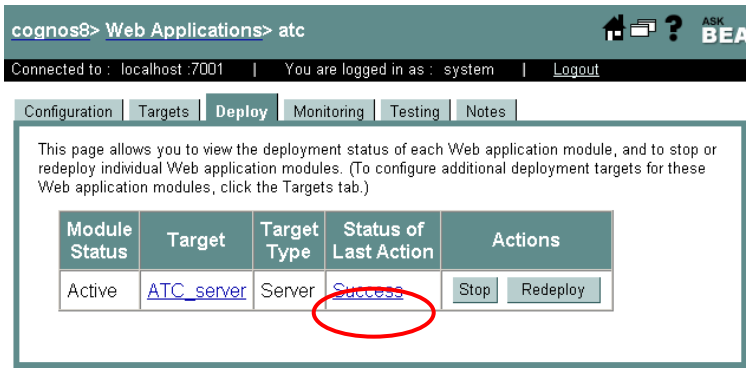


5. Select the "I will make the Web Application module accessible from the following location" option button. Under the Identify section, the default name of the application is p2pd. Change this to atc. Each application must be identified with a unique name. This does not change the context root of the application. Click the Deploy button to continue.



Note: If you use the default option "Copy this Web Application module onto every target for me", the application will be copied to the managed server target. If you choose to use this option, you don't need to extract the p2pd.war file into a separate location. However, you will not be able to set the Servlet Reload Check Secs period property for performance enhancement.

6. The Deploy option should occur automatically if not click the Deploy button. This may take a few minutes. Once the deployment has completed, the status should be "Success".

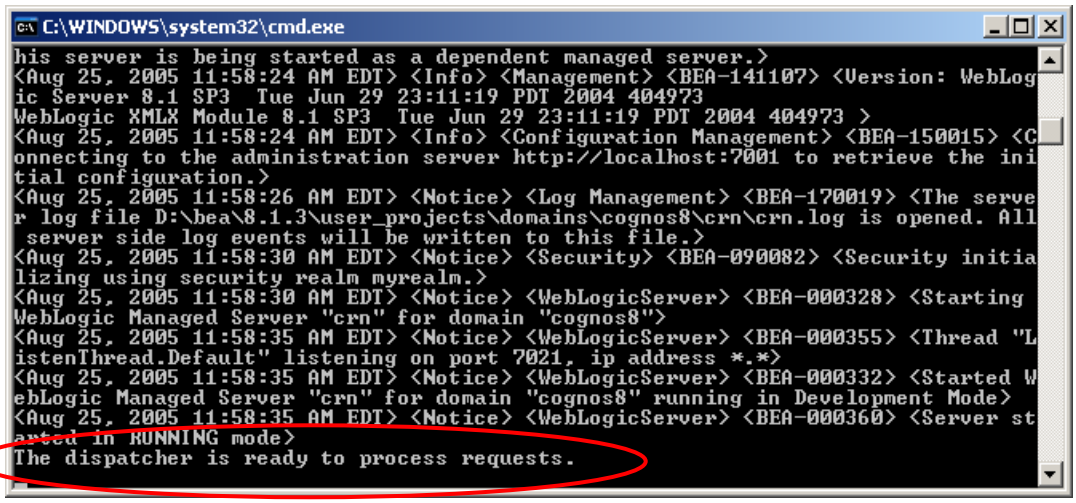


4.3.3 Post Deployment Configuration

The post deployment steps are the same as those used for the single server installation.

4.3.4 Verify the Deployed Application

Once the deployment has completed successfully, you should see a message stating "The dispatcher is ready to process requests" in the Managed Server's console. This indicates that the Dispatcher for the IBM Cognos 8 Application Tier Components has been registered to CM and has started. You should see an active BIBusTKServerMain process running when consulting the active processes on the machine. Use the appropriate utility for your OS to verify this.

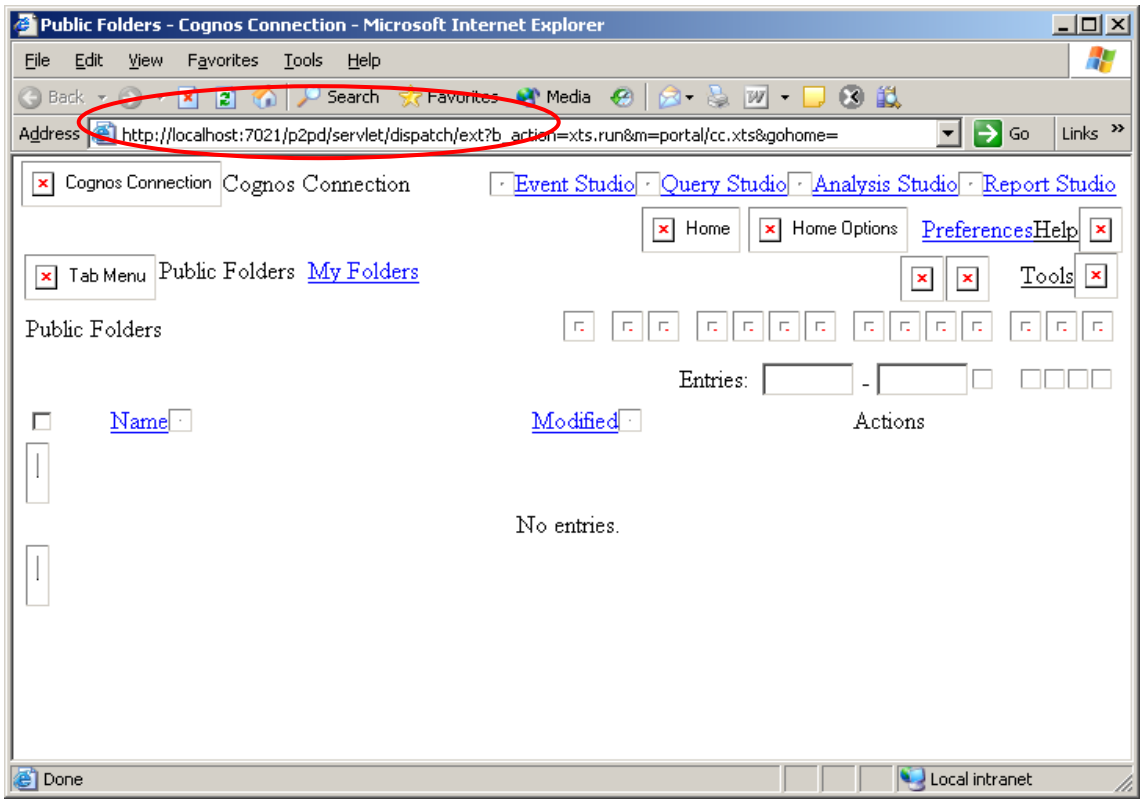


Note: The server console output to the screen is only available when the server instances are started using the scripts. They are not available if the server instances are started via the Administration Console using Node Manager.

You can check the IBM Cognos 8 logs to verify that the Dispatcher has been registered and has started.

Another test to verify that the Dispatcher has started is to connect directly to the Dispatcher via a browser. Open a browser and type in the Dispatcher URI. If you used p2pd as the context root, the default is

http://localhost:<port_number>/p2pd/servlet/dispatch/ext. In this example, the address is http://localhost:7021/p2pd/servlet/dispatch/ext. You should see the Public Folders page. If you did not include the static webcontent when creating the application in the Build Application Wizard, the page will appear incomplete. This is normal.



4.4 Servlet Gateway

You should have the Gateway (GW) installed in a separate location. The GW component is the last component to be configured and deployed once CM and ATC have been configured and deployed. You cannot configure ATC until CM is deployed and running, and you cannot not configure GW until ATC has been configured and deployed.

If you would prefer to use a web server instead of the Servlet Gateway, follow the steps outline in the IBM Cognos 8 user documentation.

4.4.1 Pre-deployment Configuration

WebLogic Server should be started prior to starting the deployment. You will not be able to access the Administration Console if the Administration Server isn't running. Make sure the Managed Server instance has been created and configured.

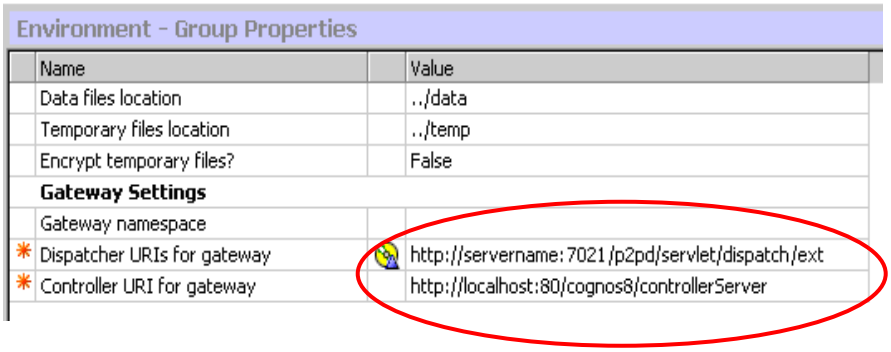
Each Cognos component must be deployed to a separate Managed Server instance for distributed scenarios. You cannot deploy two IBM Cognos applications to the same Managed Server instance. They must run on separate instances.

4.4.1.1 IBM Cognos 8 Setup

For detailed information regarding the configuration of IBM Cognos 8, refer to the Application Servers – Installing and Configuring the Cognos Application document and the IBM Cognos 8 user documentation.

The steps to configure and deploy the Servlet Gateway are similar to those for a full single-server install.

- The same environment variables need to be set.
- The JDK used when running Cognos Configuration **must** be the same as the one used for running WebLogic Server.
- The steps outlined in ASCG05 **must** be completed prior to deploying the IBM Cognos 8 application. For example, database clients must be configured and the security provider file must be copied to the appropriate location.
- The port number used by the IBM Cognos 8 Servlet Gateway is the Managed Server port that was set when the domain was created using the Domain Configuration Wizard or when the new Server was added to the domain configuration via the Administration Console. Do not use the Administration Server port. The Cognos Configuration for the Gateway component points to the Dispatcher URI for the ATC component.



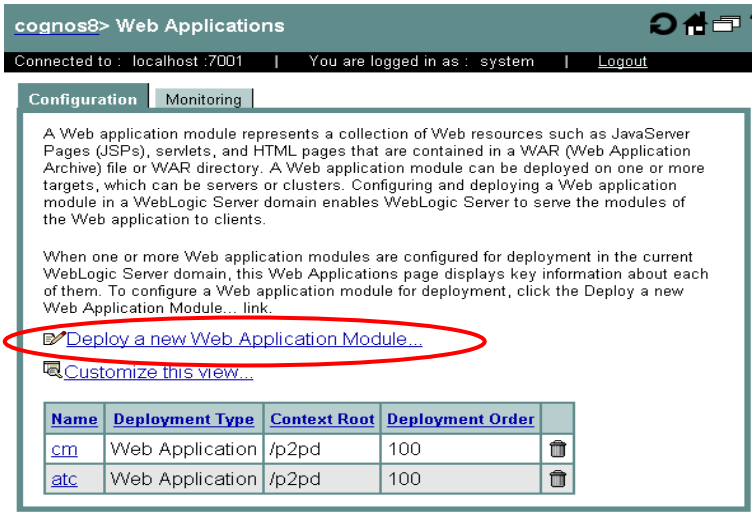
- Make sure CM and ATC are running, before configuring the Gateway using the IBM Cognos Configuration tool. They must also be running before starting the deployment for the Servlet Gateway.
- The application was created using the Build Application Wizard. You should use the Expand files into a folder option.

4.4.2 Deployment

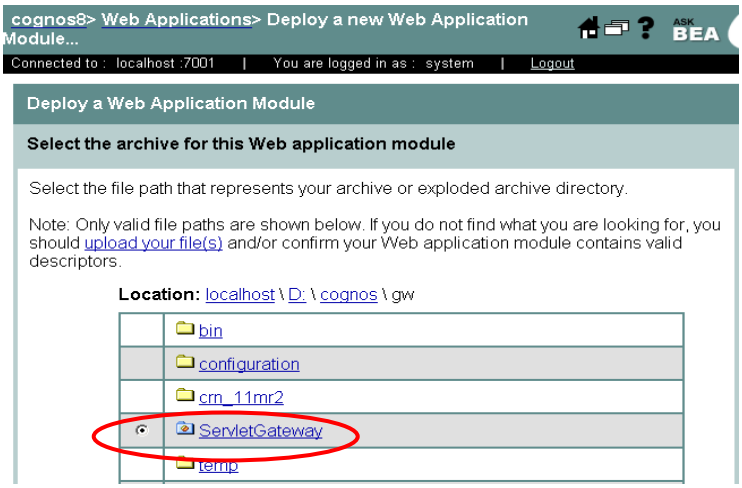
The deployment sequence in the Administration Console is the same as the steps for a single server setup. The difference here is that a deployment will occur for each component. Only the WebLogic 8.1.x dialogs will be displayed for this example. Refer to the single server steps for the WebLogic 7.0.x dialogs.

The default context root for the Servlet Gateway is ServletGateway. You can change this when creating the application in the Build Application Wizard of the IBM Cognos Configuration tool.

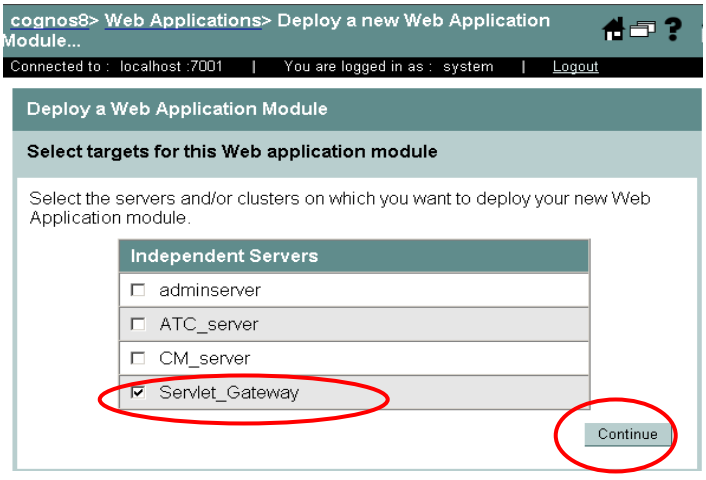
1. WebLogic Server should already be running. Start the WebLogic Console from the browser and login. The default address is http://localhost:7001/console. You can also use the server name. For example, http://servername:7001/console. User the user and password created when you created the domain.
2. Under Deployments, select the Web Applications Modules link. Click the Deploy a new Web Application Module link. You should already have the CM and ATC applications deployed and running.



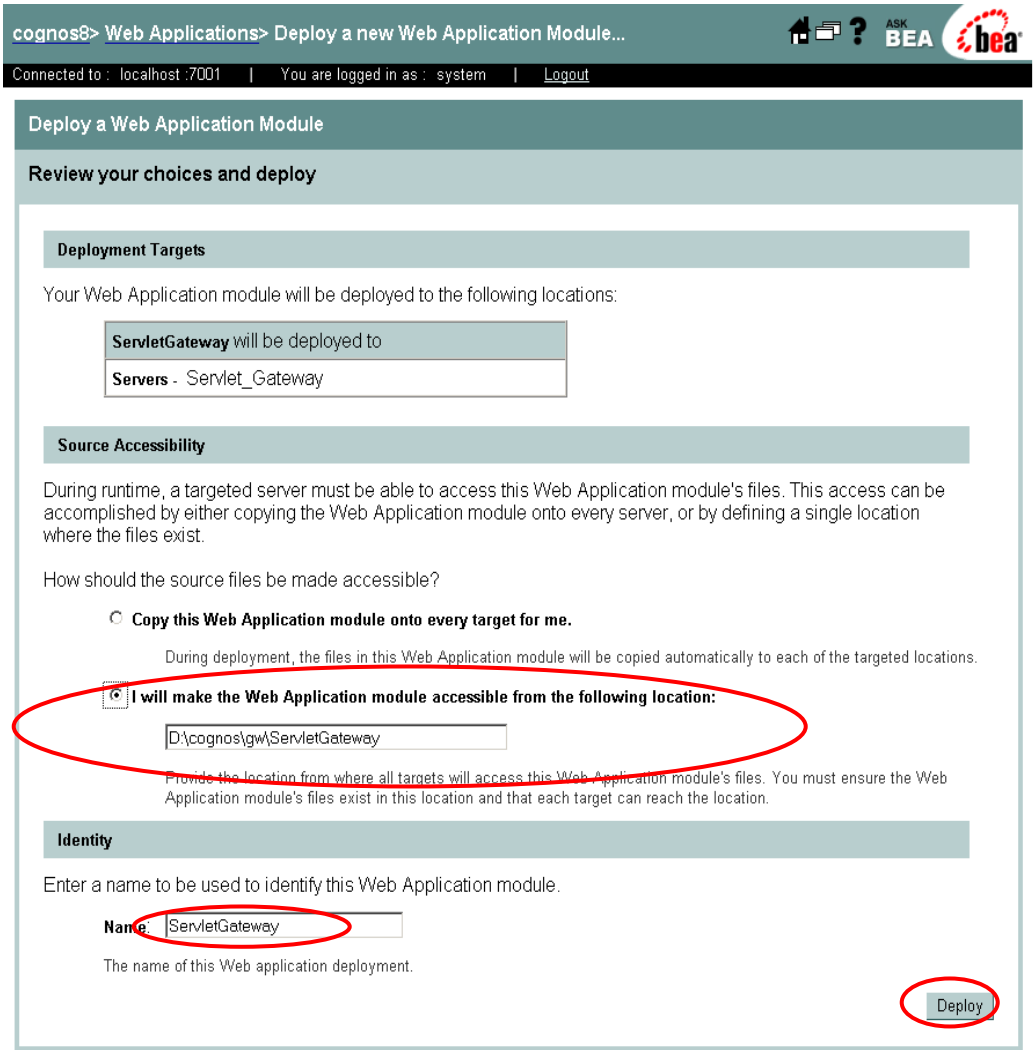
3. Browse to the ServletGateway application and select it. In this example, it is under D:\cognos\gw. Click on Target Module button.



4. Check the target server. In this example, the target server is Servlet_Gateway. Click on Continue



5. Select the "I will make the Web Application module accessible from the following location" option button. Under the Identify section, the default name of the application is ServletGateway. You can change the identifier name or leave the default. Click the Deploy button to continue.



6. The Deploy option should occur automatically if not click the Deploy button. This may take a few minutes. Once the deployment has completed, the status should be "Success".

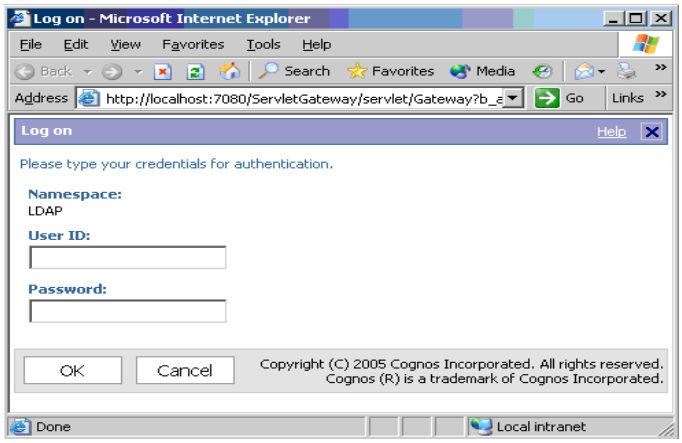
4.4.3 Post Deployment Configuration

The post deployment steps are the same as those used for the single server installation.

4.4.4 Verify the Deployed Application

Access the Portal to verify that the deployment of the Servlet Gateway was successful. Open a new browser session and enter the Gateway URI. The Gateway URI for the Servlet Gateway is http://<servername>:<port>/<context_root>/servlet/Gateway. For example, if the Context root is ServletGateway, and the managed server is on the localhost using port 7080, then the Gateway URI for the Servlet Gateway would be http://localhost:7080/ServletGateway/servlet/Gateway. IBM Cognos Connection should open and you should see the Public Folders page.

If a 3rd party namespace (eg. LDAP) was configured in Cognos Configuration, then you would be prompted with the login page.


Log on - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost:7080/ServletGateway/servlet/Gateway?b_a
Go Links
Log on
Help
Please type your credentials for authentication.
Namespace:
LDAP
User ID:
Password:
OK
Cancel
Copyright (C) 2005 Cognos Incorporated. All rights reserved.
Cognos (R) is a trademark of Cognos Incorporated.
Done
Local intranet

5 Troubleshooting

5.1 Using a WAR file

For WebLogic Server deployment, you should select the expand the application files to a folder option when using the Build Application Wizard in Cognos Configuration. However, if for some reason you cannot expand the files to a folder, you can use a WAR file. Do not use an EAR file.

The following steps use the default application name, p2pd. The context root can be changed by using a different name when creating the war file or application directory. Just remember to change this in your Cognos Configuration, as well.

1. Create the p2pd.war file using the Build Application Wizard in Cognos Configuration. Note the location of the file.
2. Create a directory called p2pd in a location accessible by the WebLogic Server. For example, you want to place the application in D:\bea\user_projects\apps. You can create the folder in Explorer or open a command prompt, navigate to D:\bea\user_projects\apps and type

   ```
   mkdir p2pd
   ```

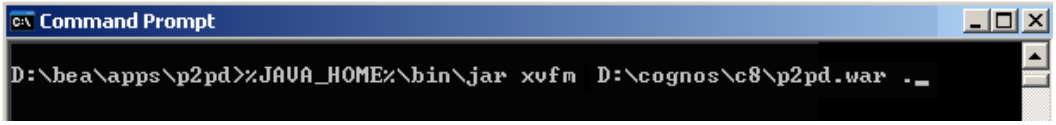
3. Extract the p2pd.war file to a directory accessible to WebLogic Server using the following command.

   Important: You must run the command from the p2pd directory created in the previous step. A space and then a period are required at the end of the command.

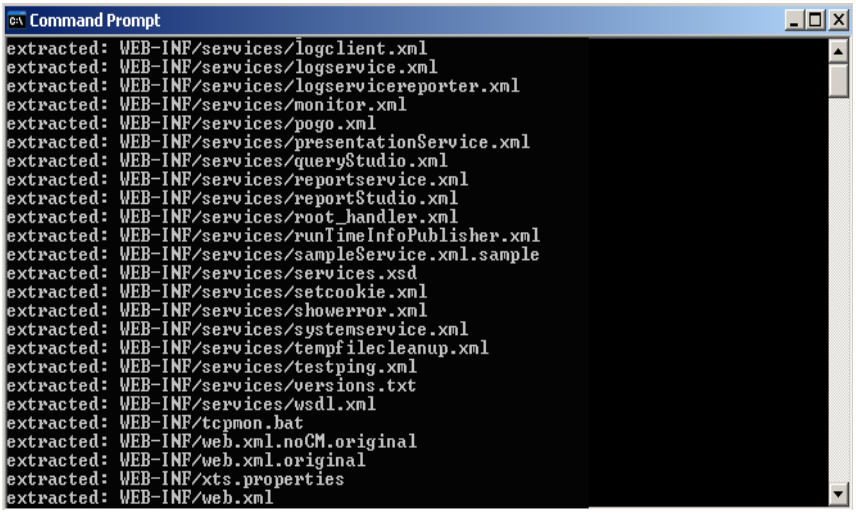
   For example,

   ```
   %JAVA_HOME%\bin\jar xvfm "<cognos8_location>/p2pd.war" .
   ```

   Note: this may take 2 to 8 minutes.



You will see that the files are extracted to a directory called WEB-INF under the p2pd directory. Once this action is complete, you can proceed with the IBM Cognos 8 deployment to WebLogic Server.



6 Application Server References / Resources

6.1 Application Server Logs

Log files for Managed Servers can be found in the domain directory structure under user_projects. The log files are named after the Managed Server instance.

- WebLogic 8.1.x:
  %WL_HOME%\user_projects\domains\<*domain_name*>\<*managed_server_name*>
- Weblogic 9.x:
  %WL_HOME%\user_projects\domains\<*domain_name*>\servers\<*managed_server_name*>\logs

6.2 Useful Web Sites

BEA's main website:
http://www.bea.com

Main page for WebLogic Server:
http://www.bea.com/framework.jsp?CNT=index.htm&FP=/content/products/weblogic/server/

BEA's Developer's Resource site:
http://dev2dev.bea.com

Main dev2dev page for WebLogic Server:
http://dev2dev.bea.com/wlserver/

List of All Supported Configurations:
http://e-docs.bea.com/platform/suppconfigs/index.html

6.3 Documentation

Online documentation can be found at the BEA website listed below. The documentation for all supported versions is available.

http://edocs.bea.com