Discovering the internals of IBM FileNet Content Platform Engine Configuration

By Praveen M Kumar

Abstract

This article provides a high level understanding of IBM FileNet P8 content management and explains the various steps involved in its configuration. Each step is explained in detail, to illustrate what the Configuration Manager tool does and why that needs to be done. While customers should always use Configuration Manager to perform these steps, we show here how the same steps could be performed manually, so that administrators may better understand the concepts in the configuration of IBM FileNet Content Platform Engine.

Author Bio: Praveen Kumar Midde is a Senior Staff Software Engineer in the IBM Enterprise Content Management group based in Hyderabad, India. He has more than eight years of experience in enterprise software development with the past six in the development group of IBM FileNet Enterprise Manager Administative tool and is currently working in the development group of IBM Case Analyzer and Case History. Praveen holds a Masters degree in Computer Science from International Institute of Information Technology (IIIT) Hyderabad, India. Reach out to him with your comments at praveen.midde@in.ibm.com

Introduction

The configuration of IBM FileNet Content Platform Engine is a simple process that is often misunderstood as a complex one. Understanding the internals of each of the configuration steps will increase understanding of the Content Platform Engine configuration process.

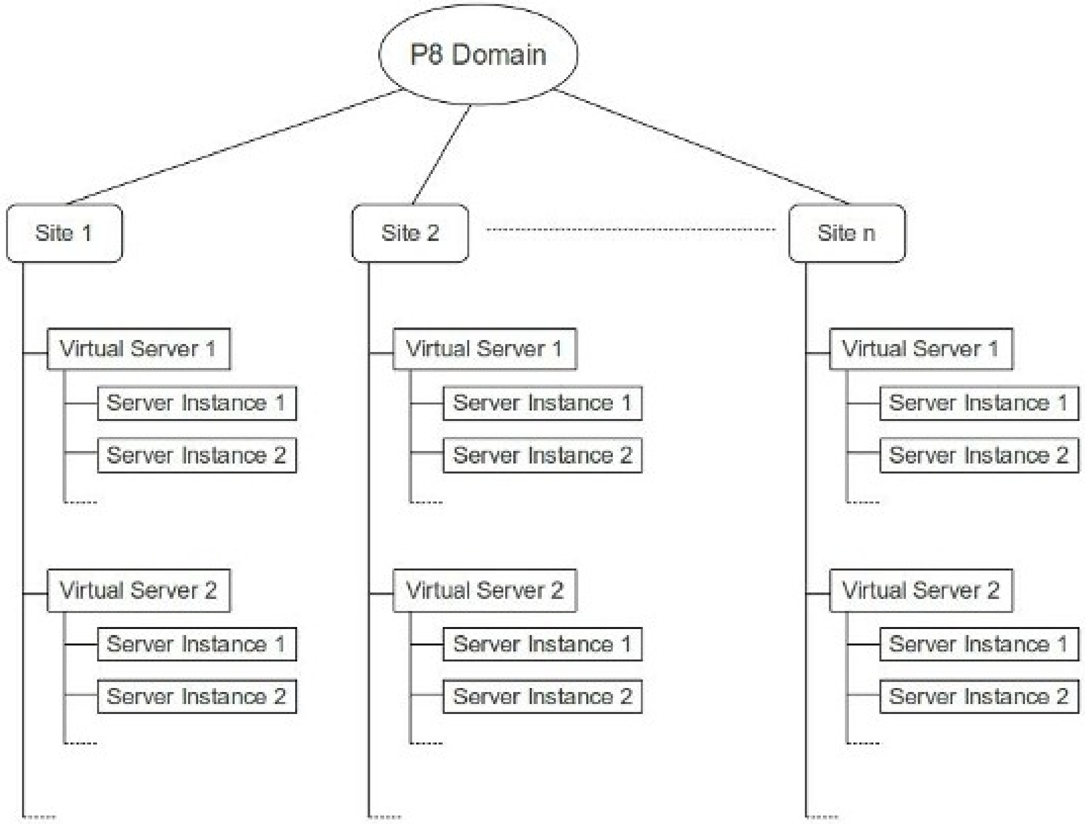
Note that configuration steps for IBM FileNet Content Manager change with each software release, and vary with the application server and database platforms being used, as well as the options selected during the configuration process. This article demonstrates manual steps that work for one application server / database / directory server / authentication environment, for one release of Content Platform Engine. Please always use the Configuration Manager tool to obtain the correct configuration settings for your release, and your environment.

P8 Content Management Overview

IBM FileNet P8 provides a platform for enterprise content management and business process management capabilities. P8 Content Platform Engine (CPE) is the core component of the P8 platform that provides the content management features.

Content Platform Engine is a J2EE-based enterprise application that is deployed in an Application server. The enterprise content is organized in various levels of hierarchy. At the root of the hierarchy is the P8 Domain, shortly referred to as just “domain”. There is only one Domain for the entire P8 platform. This domain holds all the information about the different types of “resources” under it. Under this domain, there exists Sites. A Site is just a logical grouping of different resources, mostly based on the geographic location. Under each Site, there is a Virtual Server which is best viewed as a logical grouping of Server Instances. In reality, the Virtual Server tends to map to the application server cluster and be tied to a load balancer of some sort. Each Server Instance pertains to one deployment of the CE application while the application server load balancer (logically represented by the Virtual Server) is responsible for distributing requests across the various CPE Server Instances under that Virtual Server.

Figure 1. P8 Domain hierarchy



As you see, there needs to be some place where the information of these sites, servers etc. needs to be stored. This is stored in a repository called the "Global Configuration Data" (GCD) – which, as the name suggests - stores all the configuration data of the domain. There exists one and only one GCD database; and all the CPE servers refer to this GCD database.

The main component of any enterprise is its "Content". The content and its metadata (i.e. data defining the content) is stored in an "Object Store"- which can be idealized as another database/repository. There can be multiple object stores that store information specific to different departments within the organization – like Software Development Labs department, Human Resources department etc. The information of all the object stores is stored in the global database, the GCD.

The typical documents/files in an enterprise form the "Content" and this can be stored either in the database (as that of the object store which contains the content's metadata), or in a shared file system or in third party software like IBM Image Services, EMC Centera, SnapLock etc. These third party applications are also referred to as Fixed Content Devices.

Once the content and its metadata is created inside an object stores, we need to provide security to the objects of the object stores as well as identify who can actually use the P8 domain. This falls into the category of Authentication and Authorization. Authentication talks about who can use the system; where as, Authorization talks about what operations the authenticated user can perform.

CPE doesn't itself actually define its own set of users and groups. It indeed uses the enterprise's LDAP servers which ideally store information about all the users in the enterprise. Any user of P8 must be an LDAP registry user. Whenever a user logs into P8 system (via various tools) by providing his username and password, the application server takes these credentials and verifies if the user exists in the LDAP server. If present, a security token is generated and sent to the client. The client would use this security token for any other request. This phase is called Authentication.

Only authenticated requests actually pass through the Application server to the Content Platform Engine application. The security on different objects inside the domain or an object store is defined by using "Access Control Lists" (ACLs) on that object. These ACLs are typically a matrix against the unique identifiers of the user(s)/group(s) and the permissions the user/group has on that object. Now, once the request comes to CPE, CPE would retrieve more information on the user (ex. all the various groups the user belongs to), performs a look-up on the ACLs to see if the particular user has permission to perform the operation he requested. This phase in servicing a request is called Authorization.

Overall, CPE can be visualized as an EJB application that resides inside the J2EE container, using the various services provided by the container. At a high level, CPE uses the security, transaction, database services provided by the J2EE container.

Configuring Content Platform Engine

The configuration of CPE consists of various steps that would define the interaction between the databases, LDAP servers etc. Prior to P8 4.5.0, the configuration of CPE is a tiresome and error prone. To increase the consumability of the product, a tool call "Configuration Manager" was introduced that performs the steps to configure the CPE by taking minimal user input.

The configuration of Content Platform Engine falls in five different steps as below:

1. Creating data sources

As we discussed earlier, the data and the metadata is stored in two types of databases – the Global Configuration Database (containing the information about the resources in a domain); and the Object store Database (containing the metadata of the data, and optionally the data).

CPE takes help of application server's "data source" object to talk to these databases. A data source (an object on the Application Server) typically consists of the information related to the database and also consists information on the drivers to be used to connect to this database. Note that the low-level communication to the database, management of the data sources etc. is all taken care by the Application Server.

So, you need to first inform where the GCD and the Object store database should reside. You do this by creating one set of data sources (one XA data source and non-XA data source) each for GCD and OS. The user would need to know the information about the location of the database, port number the database listener listens to, database credentials, the type and version of JDBC drivers to be used to connect to the database.

The application server tools can be used to create these data sources as described below: (Note: The example below applies specifically to DB2 for LUW. For other types of databases, please refer the product documentation.)

1.1 Database and the environment variable

Prior to creating the data source, make sure that databases are created for holding the GCD and Object store information. These databases should have the required parameters as mentioned in the FileNet guides. For example: On DB2, set the the Application heap size of the database to 2560, Statement heap size to 8192, pagesize to atleast 32K. The database user used for the data source need to have at least the “Connect to database” and “create tables” permissions.

Here is a sample script to create the DB2 database. Copy it into a file “createDB2DB.bat” and run the command `"createDB2DB DBNAME DBDRIVE DBDIR DBUSER"`

Listing 1: Sample DB2 database creation script

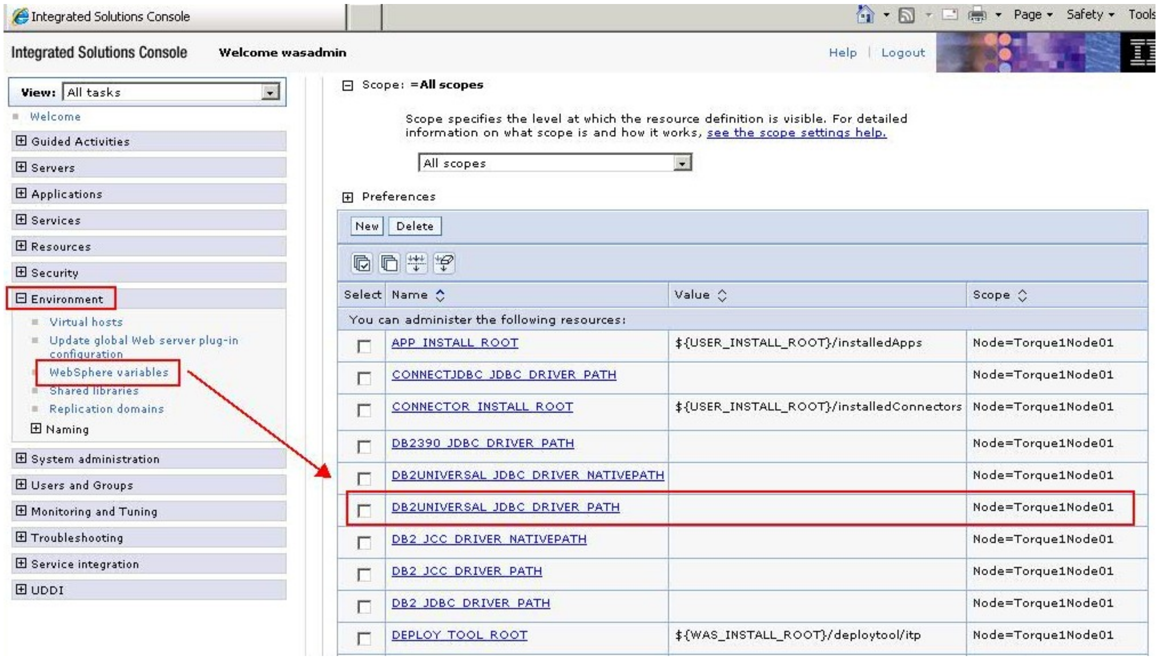
```
@echo off

REM  usage: createDB2DB DBNAME DBDRIVE DBDIR DBUSER
REM  Creates a DB2 database named DBNAME under directory DBDRIVE DBDIR
REM  accessible by user DBUSER.

if {%1}=={} goto usage
if {%2}=={} goto usage
if {%3}=={} goto usage
if {%4}=={} goto usage

set DBNAME=%1
set DBDRIVE=%2
set DBDIR=%3
set DBUSER=%4

REM  close any outstanding connections
db2 -v CONNECT RESET

REM  Go ahead and drop if it exists
db2 -v DROP DATABASE %DBNAME%

REM Create the database - these statements must be on a single line (unfortunately)
db2  -v  -t  "CREATE  DATABASE  %DBNAME%  ON  '%DBDRIVE%'  USING  CODESET  UTF-8  TERRITORY  US
COLLATE  USING  SYSTEM  CATALOG  TABLESPACE  MANAGED  BY  SYSTEM  USING  ('%DBDRIVE%%DBDIR%\
%DBNAME%\sys')  TEMPORARY  TABLESPACE  MANAGED  BY  SYSTEM  USING  ('%DBDRIVE%%DBDIR%\%DBNAME
%\systmp') USER TABLESPACE MANAGED BY SYSTEM USING ('%DBDRIVE%%DBDIR%\%DBNAME%\usr')" ;

REM  Increase the application and statement heap sizes
db2 -v UPDATE DATABASE CONFIGURATION FOR %DBNAME% USING APPLHEAPSZ 2560
db2 -v UPDATE DATABASE CONFIGURATION FOR %DBNAME% USING STMTHEAP 8192
```

```
REM  Connect to db
db2 -v CONNECT TO %DBNAME%

REM  Drop unnecessary default tablespaces
db2 -v DROP TABLESPACE USERSPACE1

REM  Create default buffer pool
db2 -v CREATE Bufferpool %DBNAME%BP IMMEDIATE SIZE -1 PAGESIZE 32K

REM  Create additional buffer pools
db2 -v CREATE Bufferpool %DBNAME%TEMPBP IMMEDIATE SIZE -1 PAGESIZE 32K
db2 -v CREATE Bufferpool %DBNAME%SYSBP IMMEDIATE SIZE -1 PAGESIZE 32K

db2 -v CONNECT RESET

db2 -v CONNECT TO %DBNAME%

REM  Create tablespaces - these statements must be on a single line
db2 -v -t "CREATE REGULAR TABLESPACE %DBNAME% PAGESIZE 32K MANAGED BY DATABASE USING
(FILE '%DBDRIVE%%DBDIR%\%DBNAME%\usr2\%DBNAME%_tbs.dbf' 512M) AUTORESIZE YES BUFFERPOOL
%DBNAME%BP" ;
db2 -v -t "CREATE USER TEMPORARY TABLESPACE USERTEMP1 PAGESIZE 32K MANAGED BY DATABASE
USING (FILE '%DBDRIVE%%DBDIR%\%DBNAME%\usrtmp\%DBNAME%_tmp.dbf' 64M) AUTORESIZE YES
BUFFERPOOL %DBNAME%TEMPBP" ;
db2 -v -t "CREATE SYSTEM TEMPORARY TABLESPACE TEMPSYS1 PAGESIZE 32K MANAGED BY SYSTEM
USING ('%DBDRIVE%%DBDIR%\%DBNAME%\systmp2') BUFFERPOOL %DBNAME%SYSBP" ;

REM  Grant USER access to tablespaces
db2 -v GRANT CREATETAB,CONNECT ON DATABASE TO user %DBUSER%
db2 -v GRANT USE OF TABLESPACE %DBNAME% TO user %DBUSER%
db2 -v GRANT USE OF TABLESPACE USERTEMP1 TO user %DBUSER%

REM  Close connection
db2 -v CONNECT RESET

goto end

:usage
@echo usage: createDB2DB DBNAME DBDRIVE DBDIR DBUSER
@echo where:
@echo        DBNAME is the name of the database (8 characters or less).
@echo        DBDRIVE is the windows driver letter followed by colon (:).
@echo        DBDIR is the existing directory that will contain the database.
@echo        This value must NOT include the drive prefix nor a trailing slash.
@echo        The database will be created in a sub-directory named [DBNAME].
@echo        DBUSER is a system user in which database access is provided.
@echo  e.g.: createDB2DB Daphne c: \DB2 db2admin
@echo        Creates a database under c:\DB2\Daphne accessible by db2admin.

:end
```

The application server uses the vendor-specific APIs (JDBC drivers) to talk to the database. In the case of WebSphere, the directory containing the library files is specified as one of the environment variables.

For ex: For DB2 database, the environment variable
“DB2UNIVERSAL_JDBC_DRIVER_PATH” can be set appropriately as below:
Navigate to Environment > WebSphere Variables >
DB2UNIVERSAL_JDBC_DRIVER_PATH

Figure 2: JDBC Driver Variable



Scope: =**All scopes**

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, see the scope settings help.

All scopes

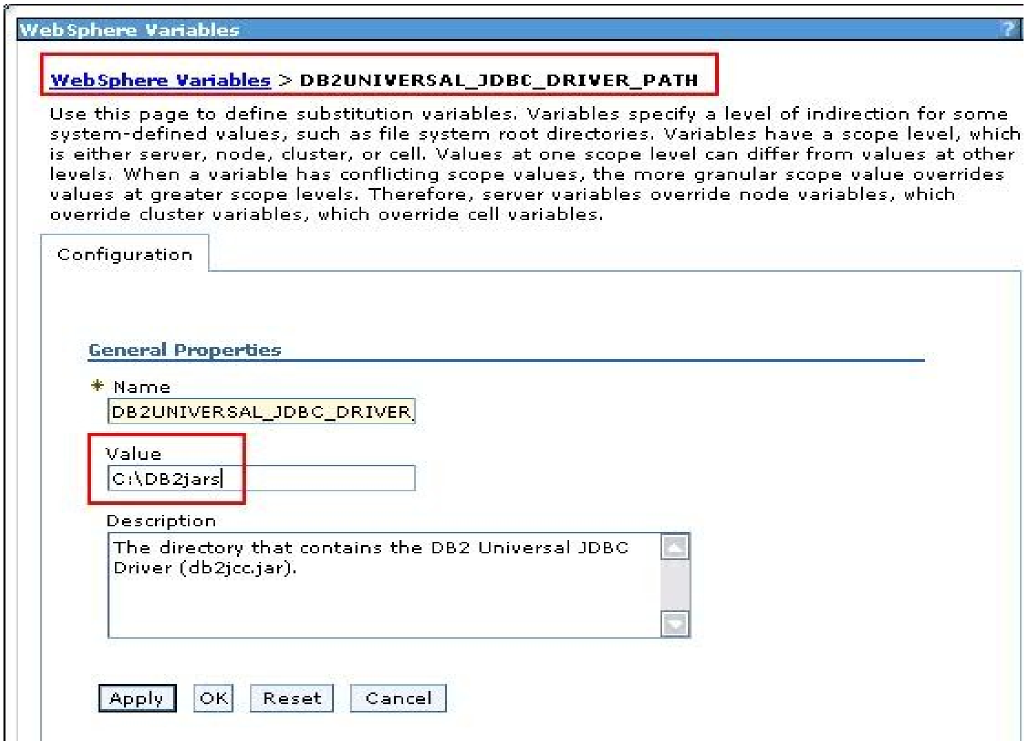
Preferences

New Delete

Select	Name	Value	Scope
You can administer the following resources:			
□	APP_INSTALL_ROOT	${USER_INSTALL_ROOT}/installedApps	Node=Torque1Node01
□	CONNECTJDBC_JDBC_DRIVER_PATH		Node=Torque1Node01
□	CONNECTOR_INSTALL_ROOT	${USER_INSTALL_ROOT}/installedConnectors	Node=Torque1Node01
□	DB2390_JDBC_DRIVER_PATH		Node=Torque1Node01
□	DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH		Node=Torque1Node01
□	DB2UNIVERSAL_JDBC_DRIVER_PATH		Node=Torque1Node01
□	DB2_JCC_DRIVER_NATIVEPATH		Node=Torque1Node01
□	DB2_JCC_DRIVER_PATH		Node=Torque1Node01
□	DB2_JDBC_DRIVER_PATH		Node=Torque1Node01
□	DEPLOY_TOOL_ROOT	${WAS_INSTALL_ROOT}/deploytool/itp	Node=Torque1Node01

Set the folder path to the required drivers (.jars) as below: (Make sure that the path doesn't contain spaces)

Figure 3: Setting the JDBC Driver Variable



WebSphere Variables > DB2UNIVERSAL_JDBC_DRIVER_PATH

Use this page to define substitution variables. Variables specify a level of indirection for some system-defined values, such as file system root directories. Variables have a scope level, which is either server, node, cluster, or cell. Values at one scope level can differ from values at other levels. When a variable has conflicting scope values, the more granular scope value overrides values at greater scope levels. Therefore, server variables override node variables, which override cluster variables, which override cell variables.

General Properties

* Name
DB2UNIVERSAL_JDBC_DRIVER

Value
C:\DB2jars

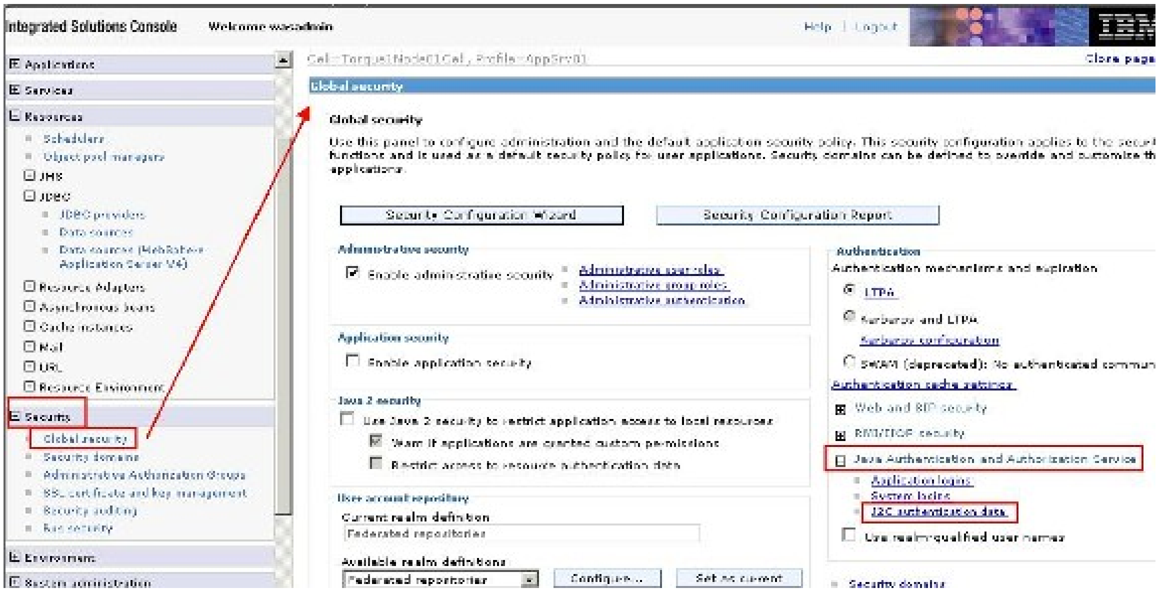
Description
The directory that contains the DB2 Universal JDBC Driver (db2jcc.jar).

1.2 Database credentials:

The database credentials to access the databases are specified using J2C authentication data specified as below:

Navigate to Security > Global Security > Java Authentication and Authorization Service > J2C Authentication data

Figure 4: J2C Authentication Data - Navigation



Cell=Torque1Node01Cell, Profile=AppSrv01

Global security

Global security

Use this panel to configure administration and the default application security policy. This security configuration applies to the securi functions and is used as a default security policy for user applications. Security domains can be defined to override and customize th applications.

Administrative security

☑ Enable administrative security

Application security

☐ Enable application security

Java 2 security

☐ Use Java 2 security to restrict application access to local resources

☑ Warn if applications are granted custom permissions

☐ Restrict access to resource authentication data

User account repository

Current realm definition

Federated repositories

Available realm definitions

Federated repositories

Authentication

Authentication mechanisms and expiration

● LTPA

○ Kerberos and LTPA

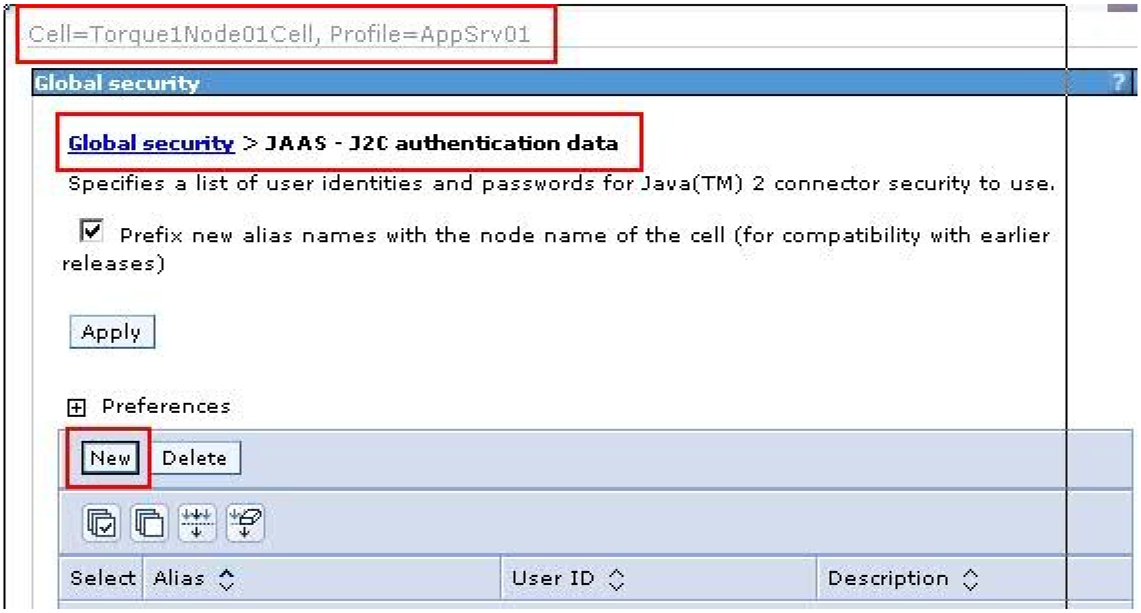
Kerberos configuration

○ SWAM (deprecated): No authenticated commun

☐ Use realm-qualified user names

Create a new Authentication alias:

Figure 5: New J2C Authentication Data – New entry



Cell=Torque1Node01Cell, Profile=AppSrv01

Global security

Global security > JAAS - J2C authentication data

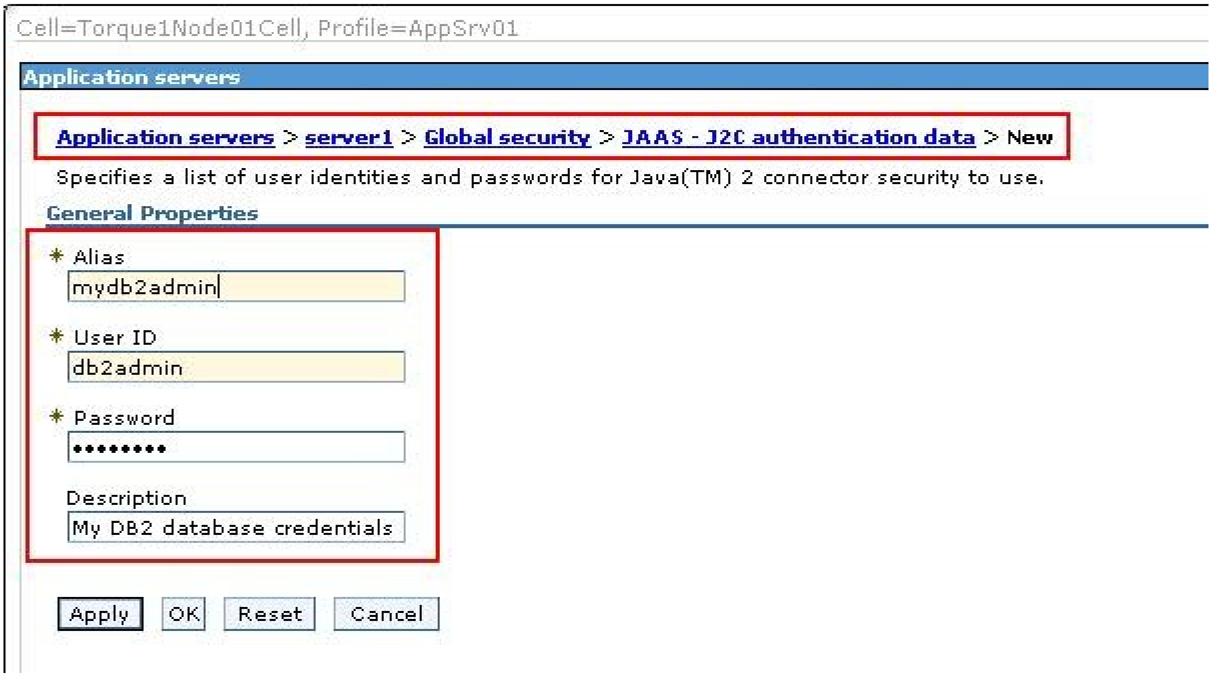
Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

☑ Prefix new alias names with the node name of the cell (for compatibility with earlier releases)

Select	Alias	User ID	Description

Provide the credentials for the database

Figure 6: J2C Authentication Data – specify database credentials



Cell=Torque1Node01Cell, Profile=AppSrv01

Application servers

Application servers > server1 > Global security > JAAS - J2C authentication data > New

Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.

General Properties

* Alias: mydb2admin

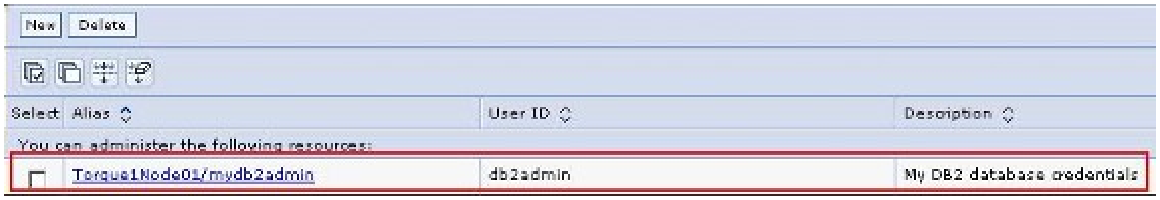
* User ID: db2admin

* Password: ●●●●●●●●●

Description: My DB2 database credentials

Once created, the new J2C Authentication alias is displayed as below:

Figure 7: J2C Authentication Data – entry after creation

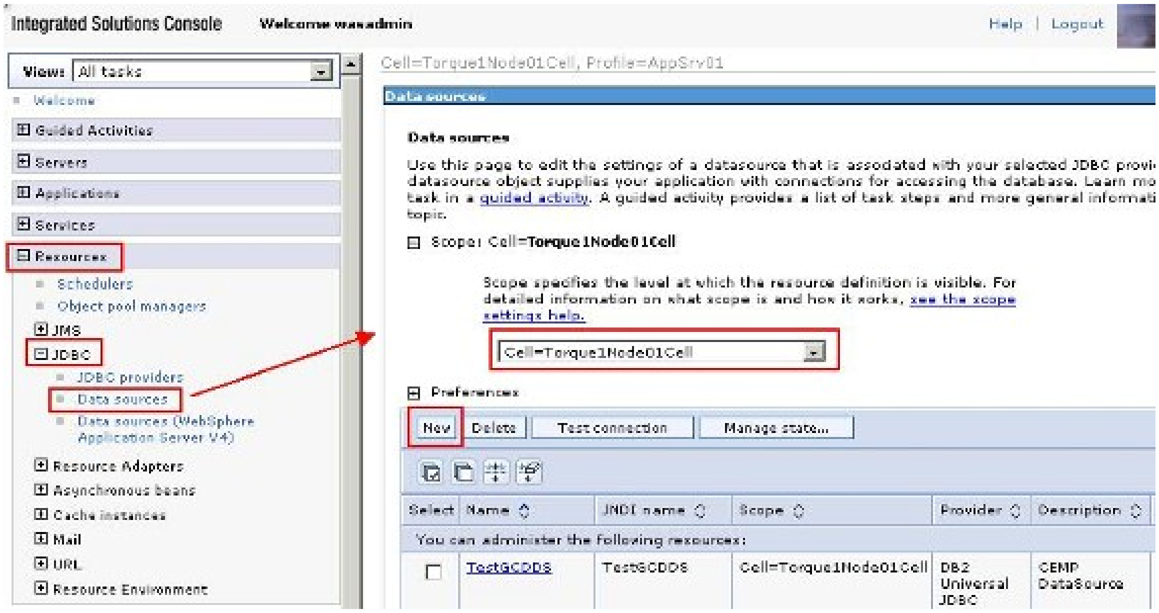


Select	Alias	User ID	Description
You can administer the following resources:			
□	Torque1Node01/mydb2admin	db2admin	My DB2 database credentials

1.3 JDBC XA and non-XA data sources

The JDBC XA and non-XA data sources are created as below:
Navigate to Resources > JDBC > Data sources
Under the Cell scope, create a New data source by clicking “New”

Figure 8: JDBC non-XA data source - Navigation



Cell=Torque1Node01Cell, Profile=AppSrv01

Data sources

Use this page to edit the settings of a datasource that is assodated with your selected JDBC provi datasource object supplies your application with connections for accessing the database. Learn mo task in a guided activity. A guided activity provides a list of task steps and more general informati topic.

⊟ Scope: Cell=Torque1Node01Cell

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, see the scope settings help.

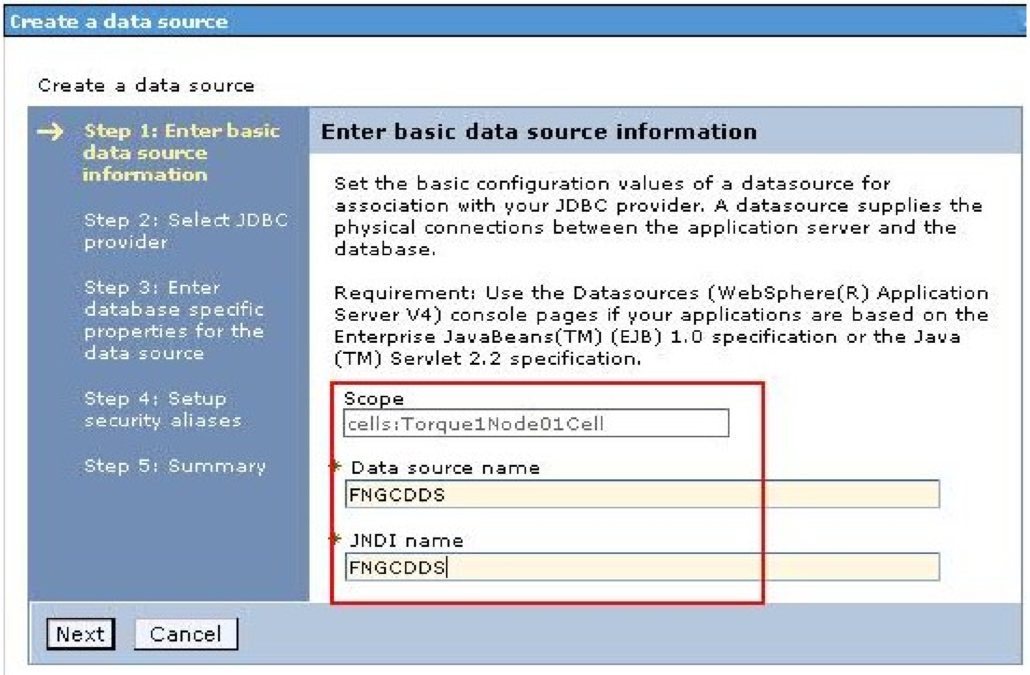
Cell=Torque1Node01Cell

⊞ Preferences

Select	Name ◇	JNDI name ◇	Scope ◇	Provider ◇	Description ◇
You can administer the following resources:					
□	TestGCDDS	TestGCDDS	Cell=Torque1Node01Cell	DB2 Universal JDBC	CEMP DataSource

Follow the wizard providing the required details for the data source creation as below:

Figure 9: JDBC non-XA data source: Name



Create a data source

→ Step 1: Enter basic data source information

Step 2: Select JDBC provider

Step 3: Enter database specific properties for the data source

Step 4: Setup security aliases

Step 5: Summary

Enter basic data source information

Set the basic configuration values of a datasource for association with your JDBC provider. A datasource supplies the physical connections between the application server and the database.

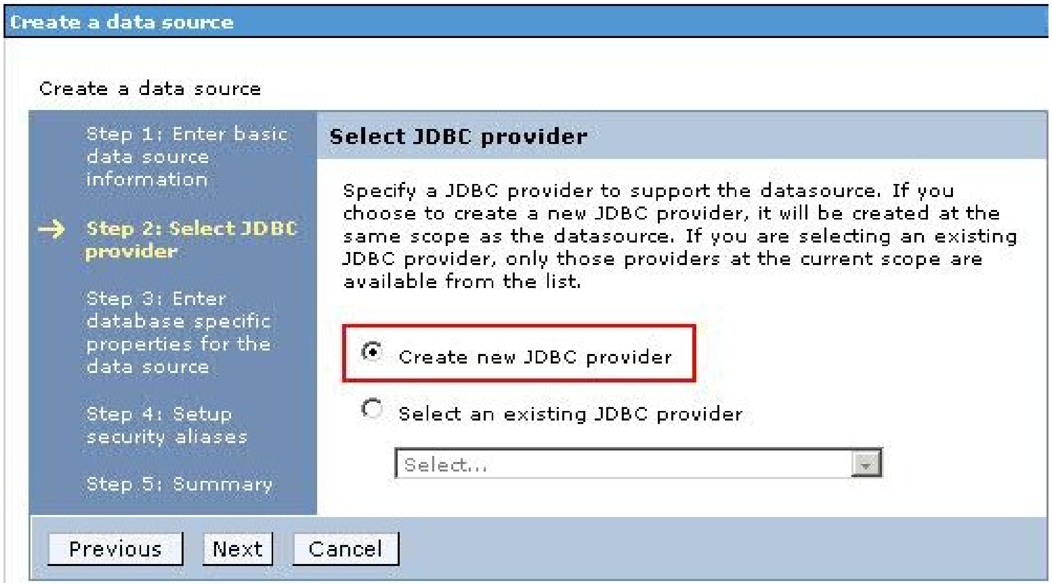
Requirement: Use the Datasources (WebSphere(R) Application Server V4) console pages if your applications are based on the Enterprise JavaBeans(TM) (EJB) 1.0 specification or the Java (TM) Servlet 2.2 specification.

Scope
cells:Torque1Node01Cell

* Data source name
FNGCDDS

* JNDI name
FNGCDDS

Figure 10: JDBC non-XA data source: New JDBC provider creation



Create a data source

Create a data source

Step 1: Enter basic data source information

→ Step 2: Select JDBC provider

Step 3: Enter database specific properties for the data source

Step 4: Setup security aliases

Step 5: Summary

Select JDBC provider

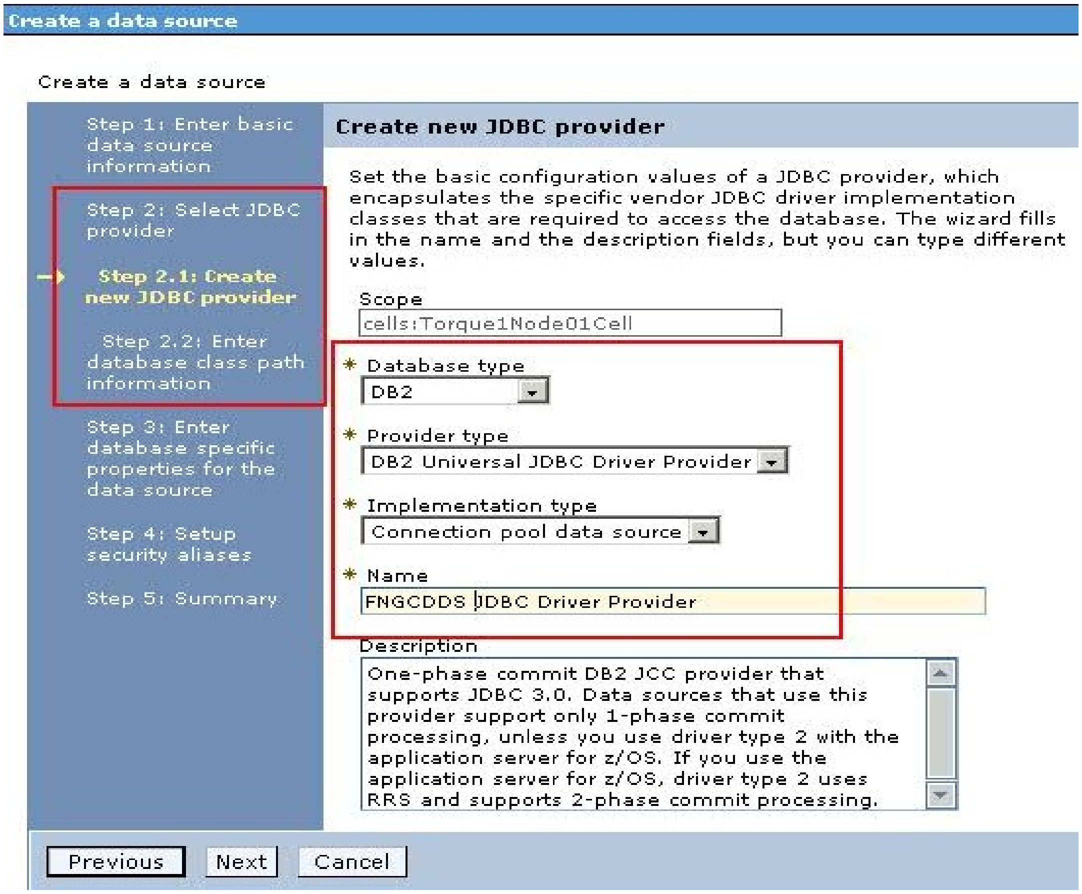
Specify a JDBC provider to support the datasource. If you choose to create a new JDBC provider, it will be created at the same scope as the datasource. If you are selecting an existing JDBC provider, only those providers at the current scope are available from the list.

● Create new JDBC provider

○ Select an existing JDBC provider

Select...

Figure 11: JDBC non-XA data source: JDBC provider details



Create a data source

Create a data source

Step 1: Enter basic data source information

Step 2: Select JDBC provider

→ Step 2.1: Create new JDBC provider

Step 2.2: Enter database class path information

Step 3: Enter database specific properties for the data source

Step 4: Setup security aliases

Step 5: Summary

Create new JDBC provider

Set the basic configuration values of a JDBC provider, which encapsulates the specific vendor JDBC driver implementation classes that are required to access the database. The wizard fills in the name and the description fields, but you can type different values.

Scope
cells:Torque1Node01Cell

* Database type
DB2

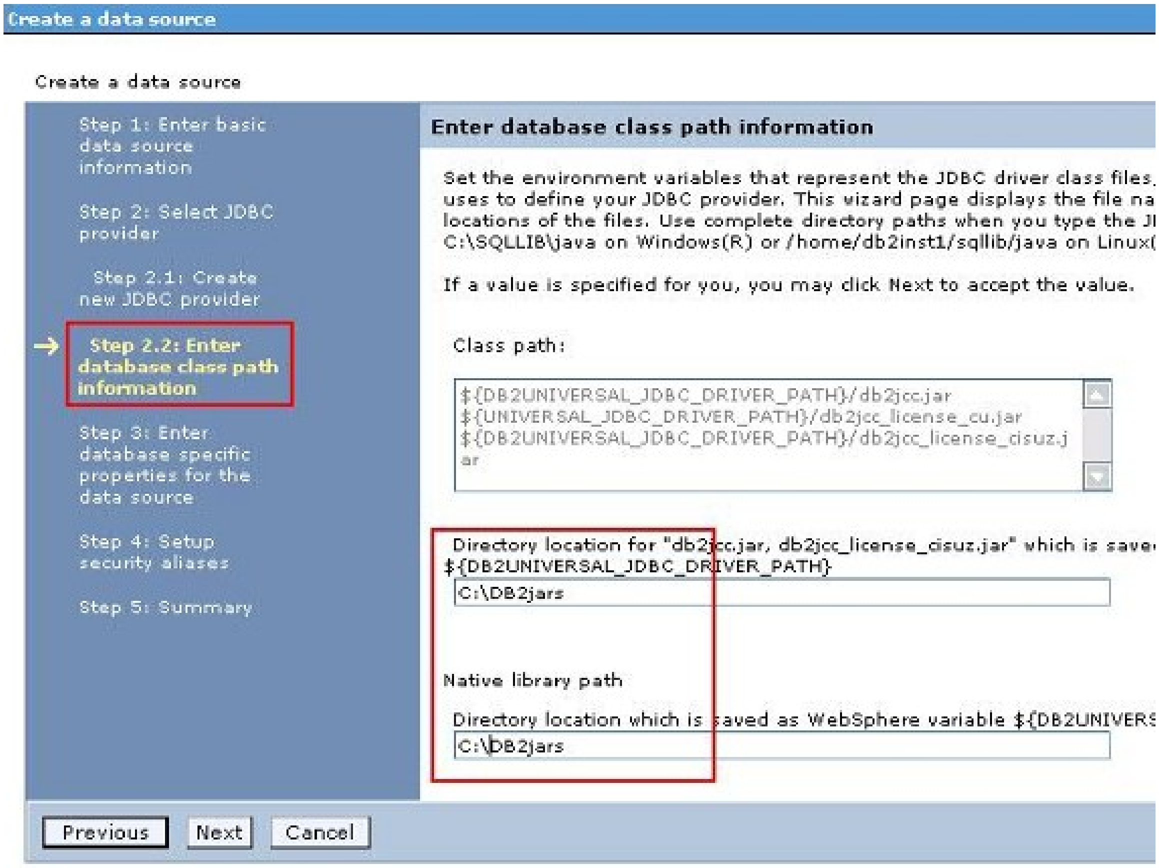
* Provider type
DB2 Universal JDBC Driver Provider

* Implementation type
Connection pool data source

* Name
FNGCDDS JDBC Driver Provider

Description
One-phase commit DB2 JCC provider that supports JDBC 3.0. Data sources that use this provider support only 1-phase commit processing, unless you use driver type 2 with the application server for z/OS. If you use the application server for z/OS, driver type 2 uses RRS and supports 2-phase commit processing.

Figure 12: JDBC non-XA data source: database class path


Create a data source
Create a data source
Step 1: Enter basic data source information
Step 2: Select JDBC provider
Step 2.1: Create new JDBC provider
Step 2.2: Enter database class path information
Step 3: Enter database specific properties for the data source
Step 4: Setup security aliases
Step 5: Summary
Previous
Next
Cancel

Enter database class path information

Set the environment variables that represent the JDBC driver class files, uses to define your JDBC provider. This wizard page displays the file na locations of the files. Use complete directory paths when you type the JI C:\SQLLIB\java on Windows(R) or /home/db2inst1/sqllib/java on Linux(

If a value is specified for you, you may click Next to accept the value.

Class path:

```
${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc.jar
${UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cu.jar
${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cisuz.j
ar
```

Directory location for "db2jcc.jar, db2jcc_license_cisuz.jar" which is save ${DB2UNIVERSAL_JDBC_DRIVER_PATH}

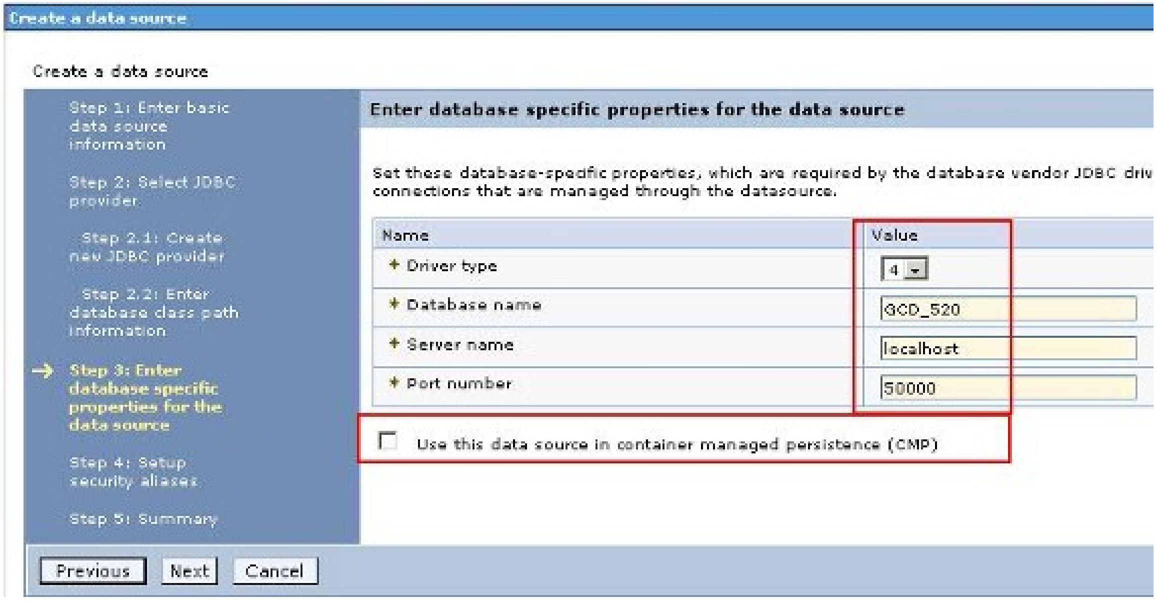
C:\DB2jars

Native library path

Directory location which is saved as WebSphere variable ${DB2UNIVERS

C:\DB2jars

Figure 13: JDBC non-XA data source: database details


Create a data source
Create a data source
Step 1: Enter basic data source information
Step 2: Select JDBC provider
Step 2.1: Create new JDBC provider
Step 2.2: Enter database class path information
Step 3: Enter database specific properties for the data source
Step 4: Setup security aliases
Step 5: Summary
Previous
Next
Cancel

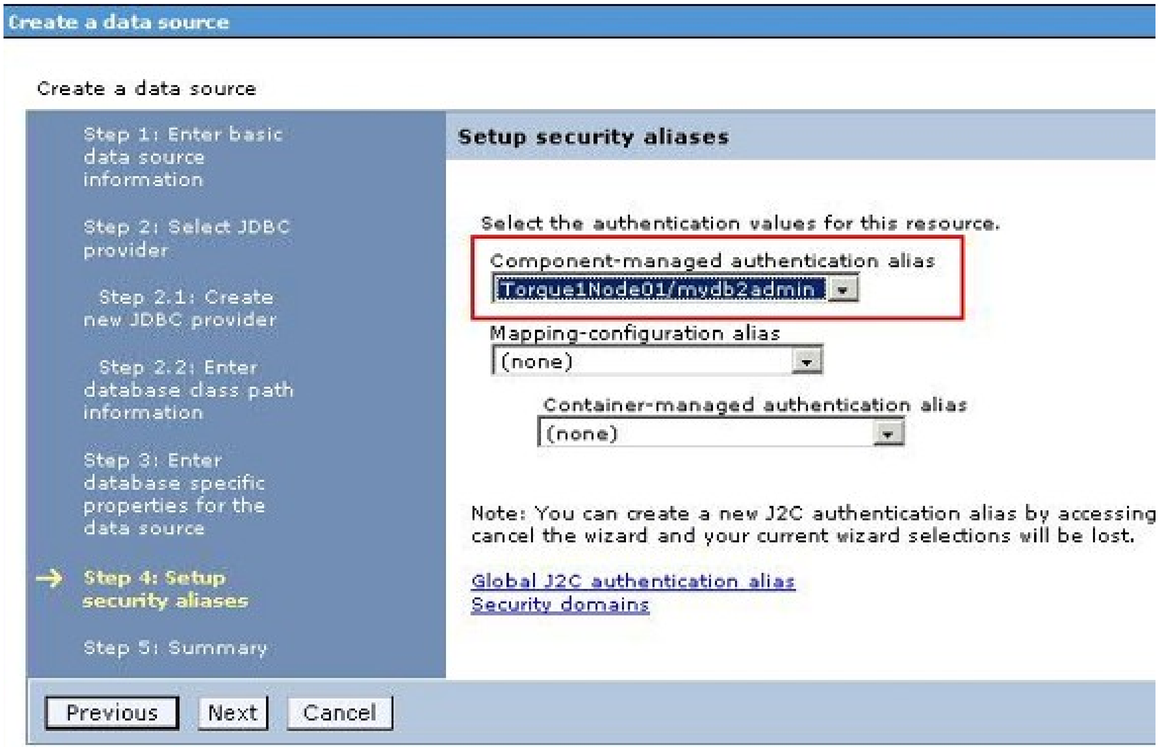
Enter database specific properties for the data source

Set these database-specific properties, which are required by the database vendor JDBC driv connections that are managed through the datasource.

Name	Value
* Driver type	4
* Database name	GCD_520
* Server name	localhost
* Port number	50000

□ Use this data source in container managed persistence (CMP)

Figure 14: JDBC non-XA data source: Authentication alias


Create a data source
Create a data source
Step 1: Enter basic data source information
Step 2: Select JDBC provider
Step 2.1: Create new JDBC provider
Step 2.2: Enter database class path information
Step 3: Enter database specific properties for the data source
→ Step 4: Setup security aliases
Step 5: Summary
Previous
Next
Cancel

Setup security aliases

Select the authentication values for this resource.

Component-managed authentication alias: Torque1Node01/mydb2admin

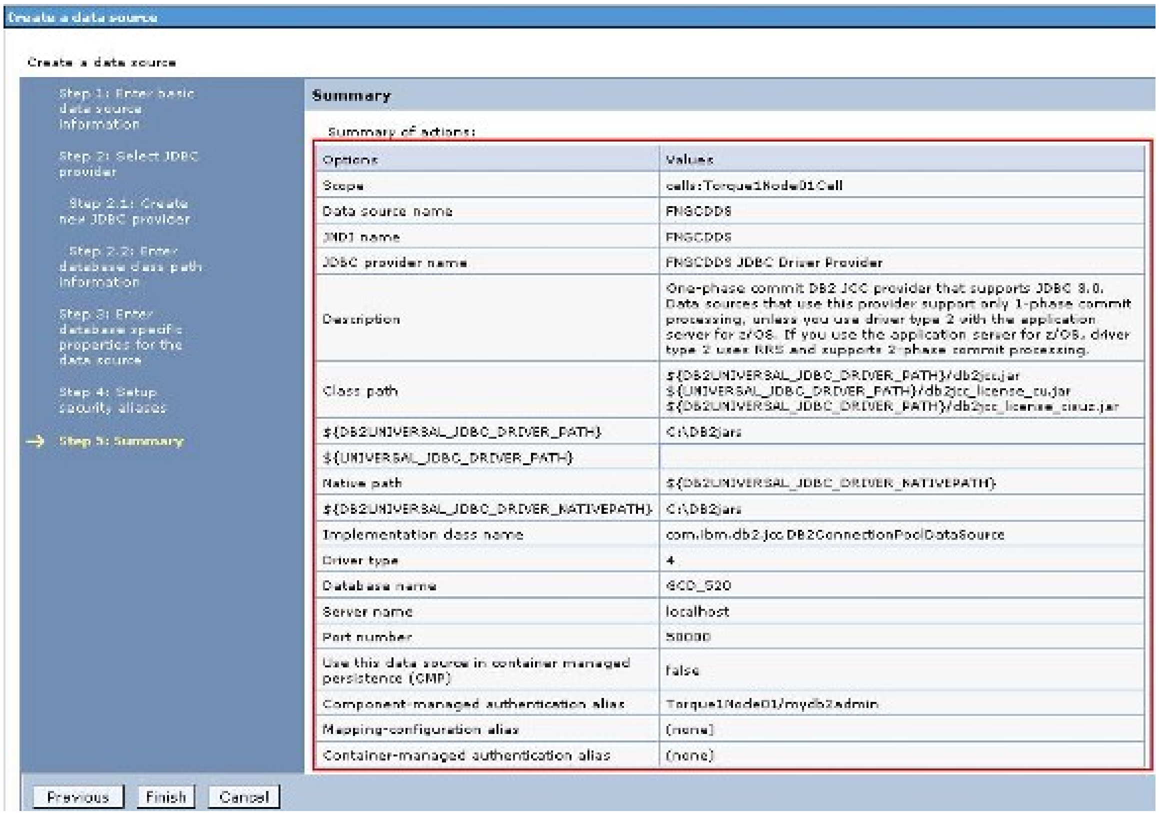
Mapping-configuration alias: (none)

Container-managed authentication alias: (none)

Note: You can create a new J2C authentication alias by accessing
cancel the wizard and your current wizard selections will be lost.

Global J2C authentication alias
Security domains

Figure 15: JDBC non-XA data source: Summary


Create a data source
Create a data source
Step 1: Enter basic data source information
Step 2: Select JDBC provider
Step 2.1: Create new JDBC provider
Step 2.2: Enter database class path information
Step 3: Enter database specific properties for the data source
Step 4: Setup security aliases
→ Step 5: Summary
Previous
Finish
Cancel

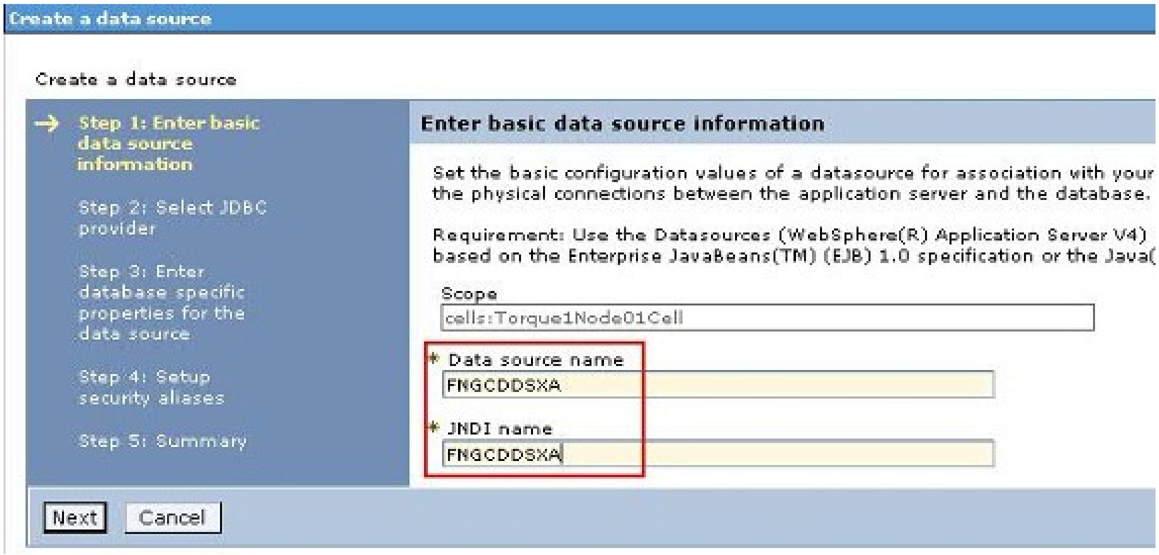
Summary

Summary of actions:

Options	Values
Scope	cells:Torque1Node01Cell
Data source name	FNGCDDS
JNDI name	FNGCDDS
JDBC provider name	FNGCDDS JDBC Driver Provider
Description	One-phase commit DB2 JCC provider that supports JDBC 3.0. Data sources that use this provider support only 1-phase commit processing, unless you use driver type 2 with the application server for z/OS. If you use the application server for z/OS, driver type 2 uses RRS and supports 2-phase commit processing.
Class path	${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc.jar ${UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cu.jar ${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cisuz.jar
${DB2UNIVERSAL_JDBC_DRIVER_PATH}	C:\DB2jars
${UNIVERSAL_JDBC_DRIVER_PATH}	
Native path	${DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH}
${DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH}	C:\DB2jars
Implementation class name	com.ibm.db2.jcc.DB2ConnectionPoolDataSource
Driver type	4
Database name	GCD_520
Server name	localhost
Port number	50000
Use this data source in container managed persistence (CMP)	false
Component-managed authentication alias	Torque1Node01/mydb2admin
Mapping-configuration alias	(none)
Container-managed authentication alias	(none)

Similarly create JDBC XA data source as below:

Figure 16: JDBC XA Data source – Name



Create a data source
Create a data source
→ Step 1: Enter basic data source information
Step 2: Select JDBC provider
Step 3: Enter database specific properties for the data source
Step 4: Setup security aliases
Step 5: Summary
Next
Cancel

Enter basic data source information

Set the basic configuration values of a datasource for association with your
the physical connections between the application server and the database.

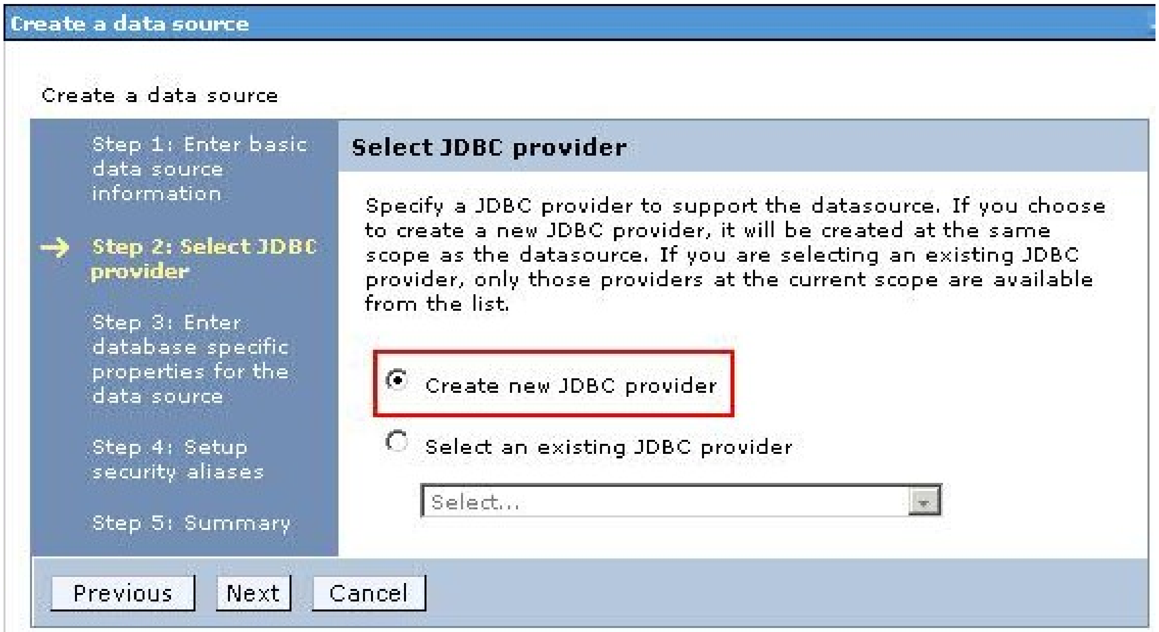
Requirement: Use the Datasources (WebSphere(R) Application Server V4)
based on the Enterprise JavaBeans(TM) (EJB) 1.0 specification or the Java(

Scope
cells:Torque1Node01Cell

* Data source name
FNGCDDSXA

* JNDI name
FNGCDDSXA

Figure 17: JDBC XA Data source – new JDBC provider



Create a data source
Create a data source
Step 1: Enter basic data source information
→ Step 2: Select JDBC provider
Step 3: Enter database specific properties for the data source
Step 4: Setup security aliases
Step 5: Summary
Previous
Next
Cancel

Select JDBC provider

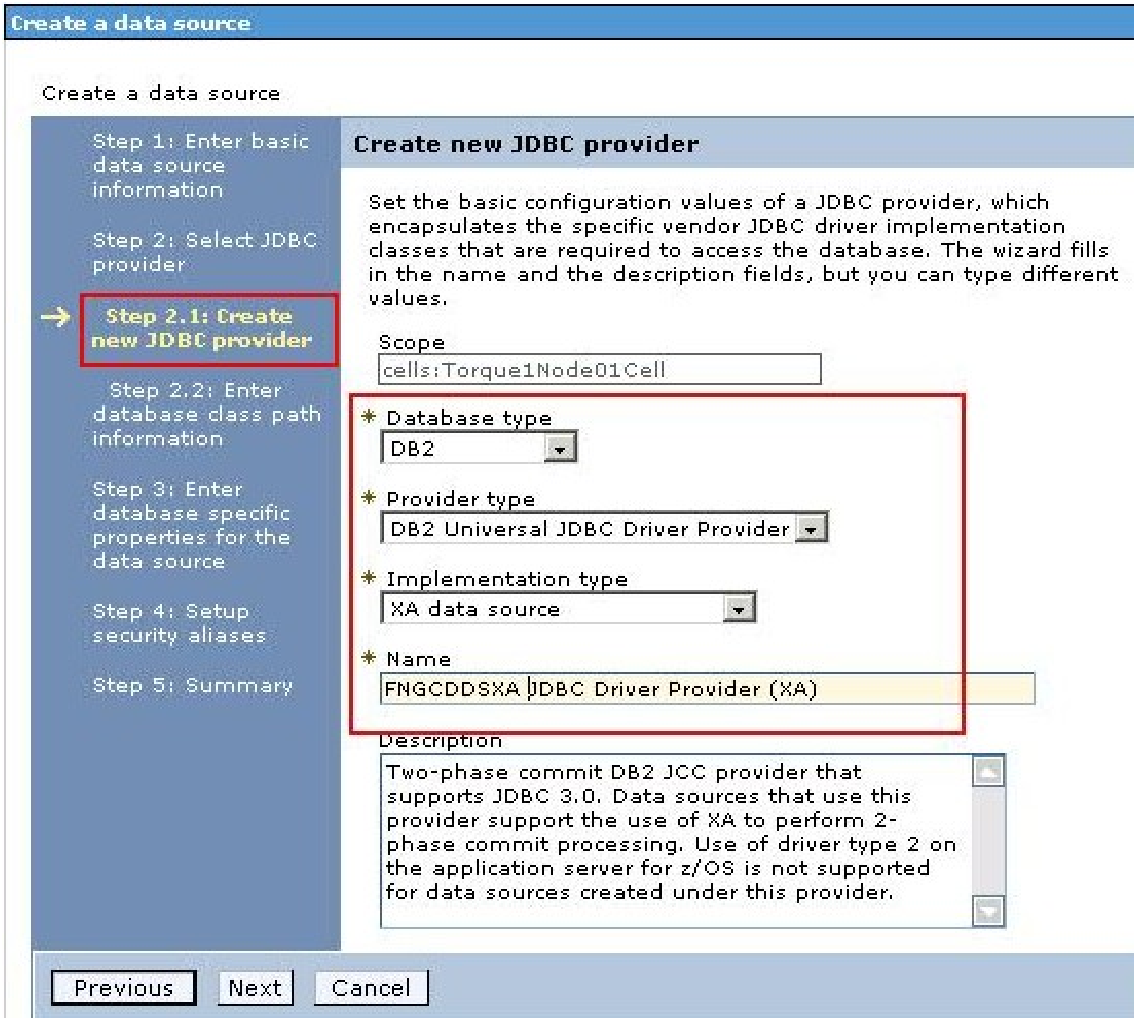
Specify a JDBC provider to support the datasource. If you choose to create a new JDBC provider, it will be created at the same scope as the datasource. If you are selecting an existing JDBC provider, only those providers at the current scope are available from the list.

● Create new JDBC provider

○ Select an existing JDBC provider

Select...

Figure 18: JDBC XA Data source – JDBC provider details



Create a data source

Create a data source

Step 1: Enter basic data source information
Step 2: Select JDBC provider
Step 2.1: Create new JDBC provider
Step 2.2: Enter database class path information
Step 3: Enter database specific properties for the data source
Step 4: Setup security aliases
Step 5: Summary

Create new JDBC provider

Set the basic configuration values of a JDBC provider, which encapsulates the specific vendor JDBC driver implementation classes that are required to access the database. The wizard fills in the name and the description fields, but you can type different values.

Scope
cells:Torque1Node01Cell

* Database type
DB2

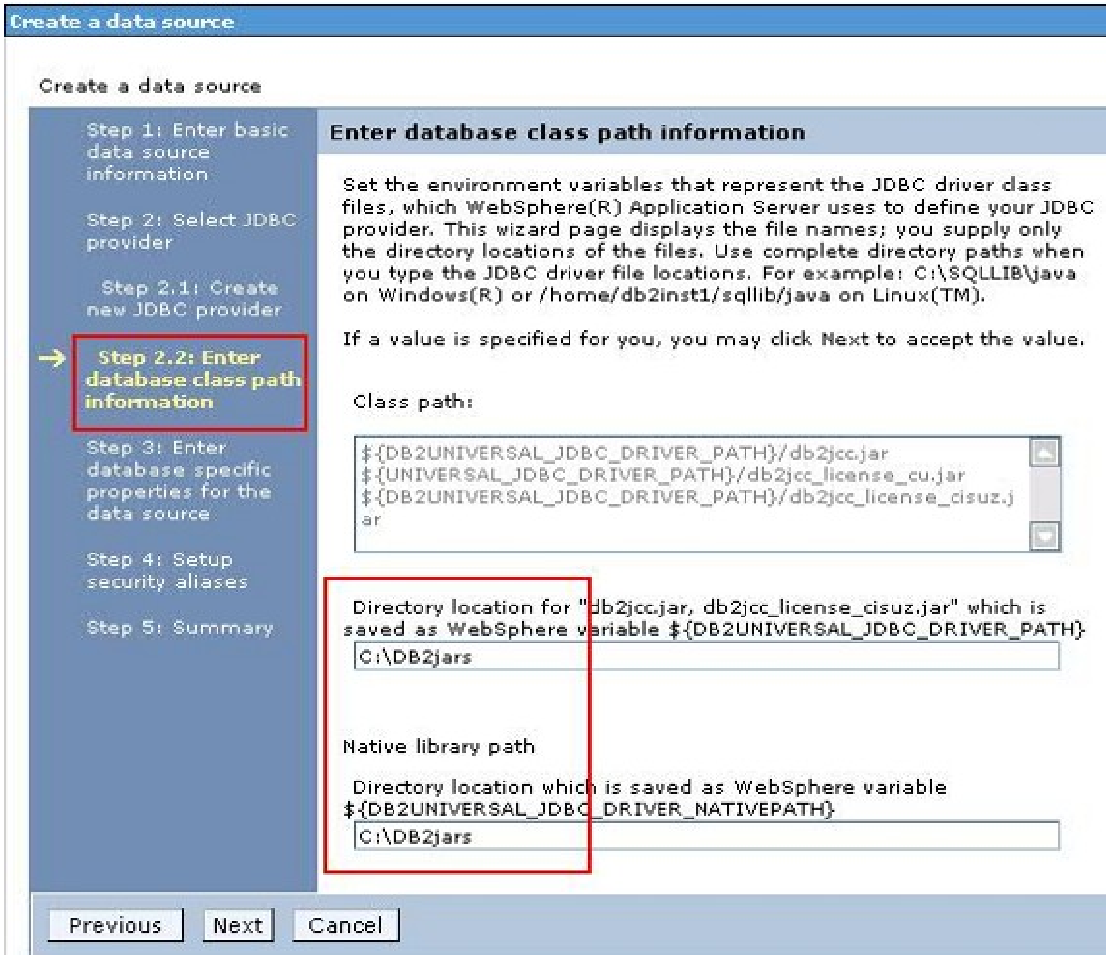
* Provider type
DB2 Universal JDBC Driver Provider

* Implementation type
XA data source

* Name
FNGCDDSXA JDBC Driver Provider (XA)

Description
Two-phase commit DB2 JCC provider that supports JDBC 3.0. Data sources that use this provider support the use of XA to perform 2-phase commit processing. Use of driver type 2 on the application server for z/OS is not supported for data sources created under this provider.

Figure 19: JDBC XA Data source – database class path information


Create a data source
Create a data source
Step 1: Enter basic data source information
Step 2: Select JDBC provider
Step 2.1: Create new JDBC provider
Step 2.2: Enter database class path information
Step 3: Enter database specific properties for the data source
Step 4: Setup security aliases
Step 5: Summary
Previous
Next
Cancel

Enter database class path information

Set the environment variables that represent the JDBC driver class files, which WebSphere(R) Application Server uses to define your JDBC provider. This wizard page displays the file names; you supply only the directory locations of the files. Use complete directory paths when you type the JDBC driver file locations. For example: C:\SQLLIB\java on Windows(R) or /home/db2inst1/sqllib/java on Linux(TM).

If a value is specified for you, you may click Next to accept the value.

Class path:

```
${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc.jar
${UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cu.jar
${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cisuz.j
ar
```

Directory location for "db2jcc.jar, db2jcc_license_cisuz.jar" which is saved as WebSphere variable ${DB2UNIVERSAL_JDBC_DRIVER_PATH}

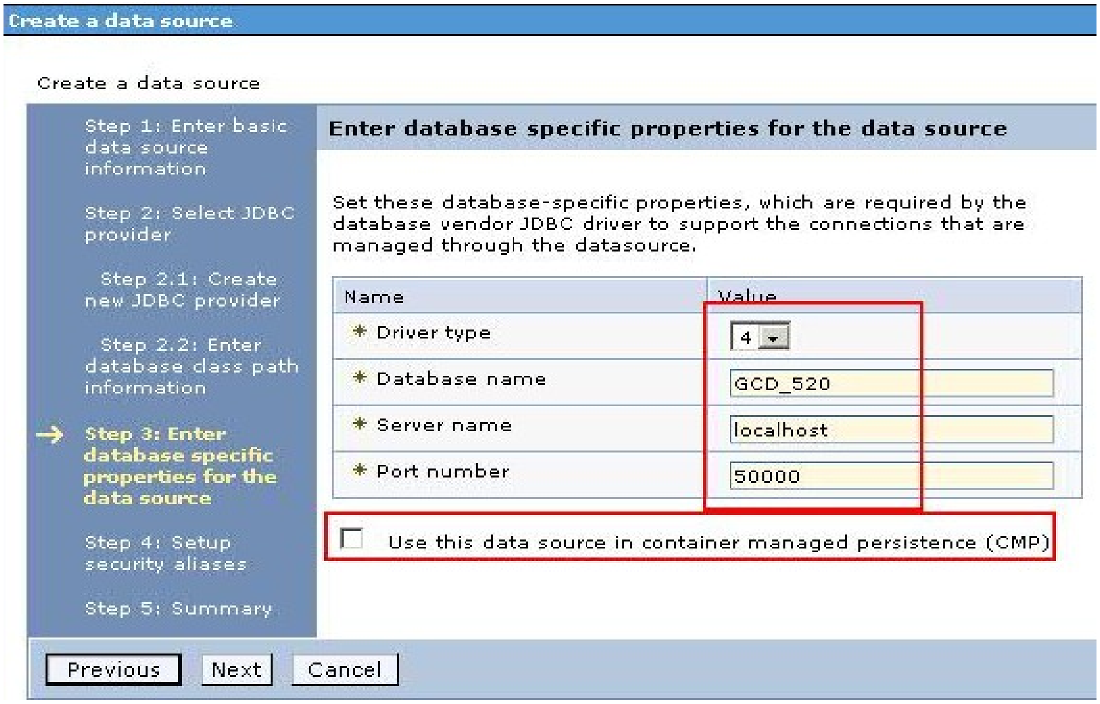
C:\DB2jars

Native library path

Directory location which is saved as WebSphere variable ${DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH}

C:\DB2jars

Figure 20: JDBC XA Data source – database specific properties


Create a data source
Step 1: Enter basic data source information
Step 2: Select JDBC provider
Step 2.1: Create new JDBC provider
Step 2.2: Enter database class path information
Step 3: Enter database specific properties for the data source
Step 4: Setup security aliases
Step 5: Summary
Previous
Next
Cancel

Create a data source

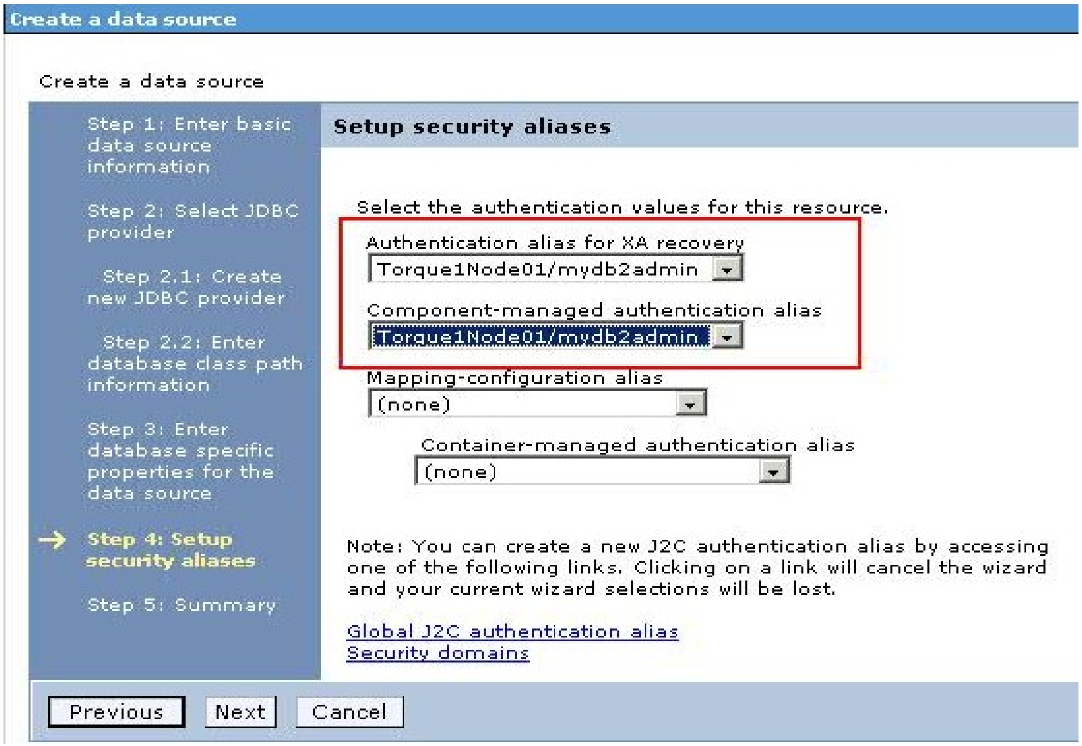
Enter database specific properties for the data source

Set these database-specific properties, which are required by the database vendor JDBC driver to support the connections that are managed through the datasource.

Name	Value
* Driver type	4
* Database name	GCD_520
* Server name	localhost
* Port number	50000

☐ Use this data source in container managed persistence (CMP)

Figure 21: JDBC XA Data source – security aliases


Create a data source
Step 1: Enter basic data source information
Step 2: Select JDBC provider
Step 2.1: Create new JDBC provider
Step 2.2: Enter database class path information
Step 3: Enter database specific properties for the data source
Step 4: Setup security aliases
Step 5: Summary
Previous
Next
Cancel

Create a data source

Setup security aliases

Select the authentication values for this resource.

Authentication alias for XA recovery: Torque1Node01/mydb2admin

Component-managed authentication alias: Torque1Node01/mydb2admin

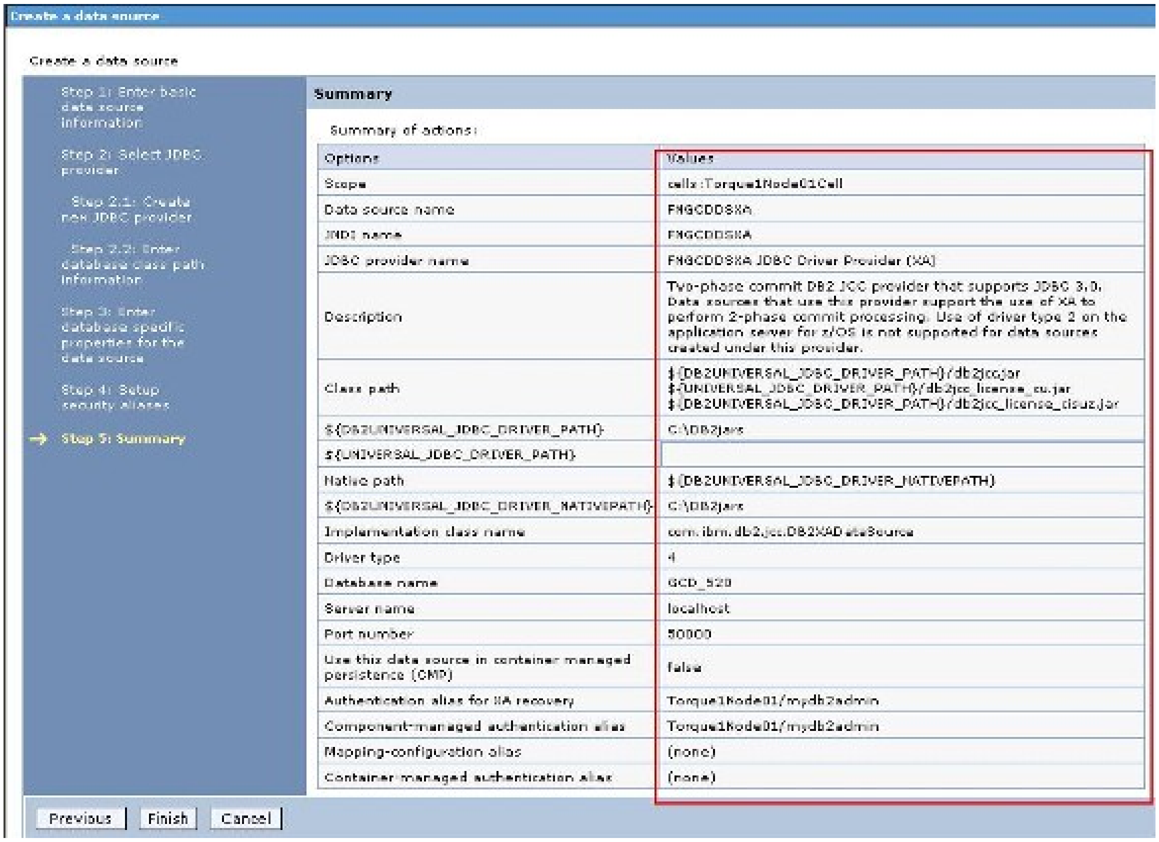
Mapping-configuration alias: (none)

Container-managed authentication alias: (none)

Note: You can create a new J2C authentication alias by accessing one of the following links. Clicking on a link will cancel the wizard and your current wizard selections will be lost.

Global J2C authentication alias
Security domains

Figure 22: JDBC XA Data source – summary



Create a data source

Create a data source

Step 1: Enter basic data source information
Step 2: Select JDBC provider
Step 2.1: Create new JDBC provider
Step 2.2: Enter database class path information
Step 3: Enter database specific properties for the data source
Step 4: Setup security aliases
→ Step 5: Summary

Summary

Summary of actions:

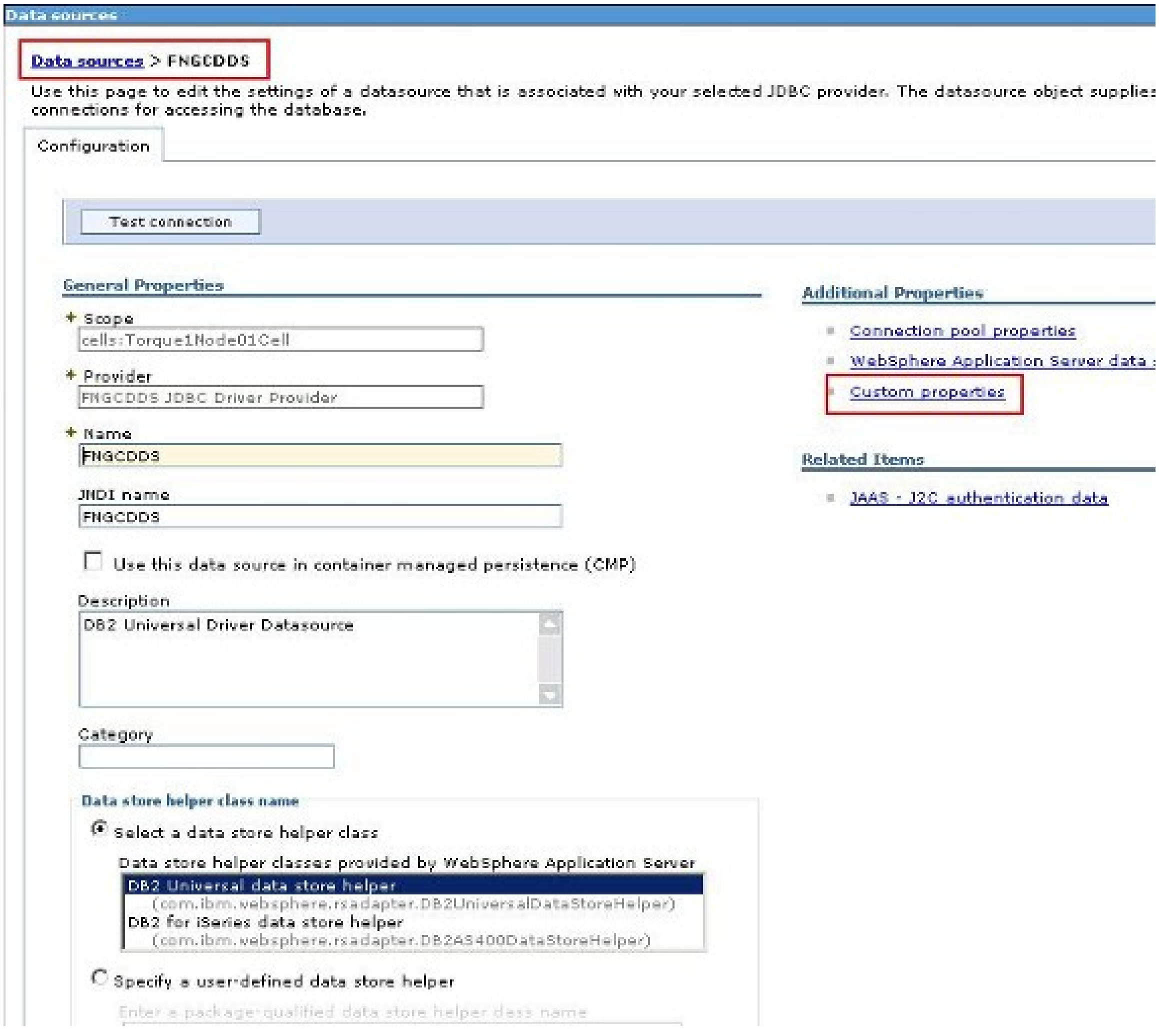
Options	Values
Scope	cells:Torque1Node01Cell
Data source name	FNGCDDSXA
JNDI name	FNGCDDSXA
JDBC provider name	FNGCDDSXA JDBC Driver Provider (XA)
Description	Two-phase commit DB2 JCC provider that supports JDBC 3.0. Data sources that use this provider support the use of XA to perform 2-phase commit processing. Use of driver type 2 on the application server for z/OS is not supported for data sources created under this provider.
Class path	${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc.jar ${UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cu.jar ${DB2UNIVERSAL_JDBC_DRIVER_PATH}/db2jcc_license_cisuz.jar
${DB2UNIVERSAL_JDBC_DRIVER_PATH}	C:\DB2jars
${UNIVERSAL_JDBC_DRIVER_PATH}	
Native path	${DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH}
${DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH}	C:\DB2jars
Implementation class name	com.ibm.db2.jcc.DB2XADataSource
Driver type	4
Database name	GCD_520
Server name	localhost
Port number	50000
Use this data source in container managed persistence (CMP)	false
Authentication alias for XA recovery	Torque1Node01/mydb2admin
Component-managed authentication alias	Torque1Node01/mydb2admin
Mapping-configuration alias	(none)
Container-managed authentication alias	(none)

1.4 Custom properties for the created data sources

For the created XA and non-XA data sources, update (or create new if these doesn't exist already) two custom properties “resultSetHoldability” and “WebSphereDefaultIsolationLevel” as below:

Navigate to the “Custom Properties” link on the respective data sources

Figure 23: Data source custom properties



Data sources

Data sources > FNGCDDS

Use this page to edit the settings of a datasource that is associated with your selected JDBC provider. The datasource object supplies connections for accessing the database.

Configuration

Test connection

General Properties

* Scope
cells:Torque1Node01Cell

* Provider
FNGCDDS JDBC Driver Provider

* Name
FNGCDDS

JNDI name
FNGCDDS

□ Use this data source in container managed persistence (CMP)

Description
DB2 Universal Driver Datasource

Category

Data store helper class name

● Select a data store helper class

Data store helper classes provided by WebSphere Application Server

DB2 Universal data store helper
(com.ibm.websphere.rsadapter.DB2UniversalDataStoreHelper)
DB2 for iSeries data store helper
(com.ibm.websphere.rsadapter.DB2AS400DataStoreHelper)

○ Specify a user-defined data store helper

Enter a package-qualified data store helper class name

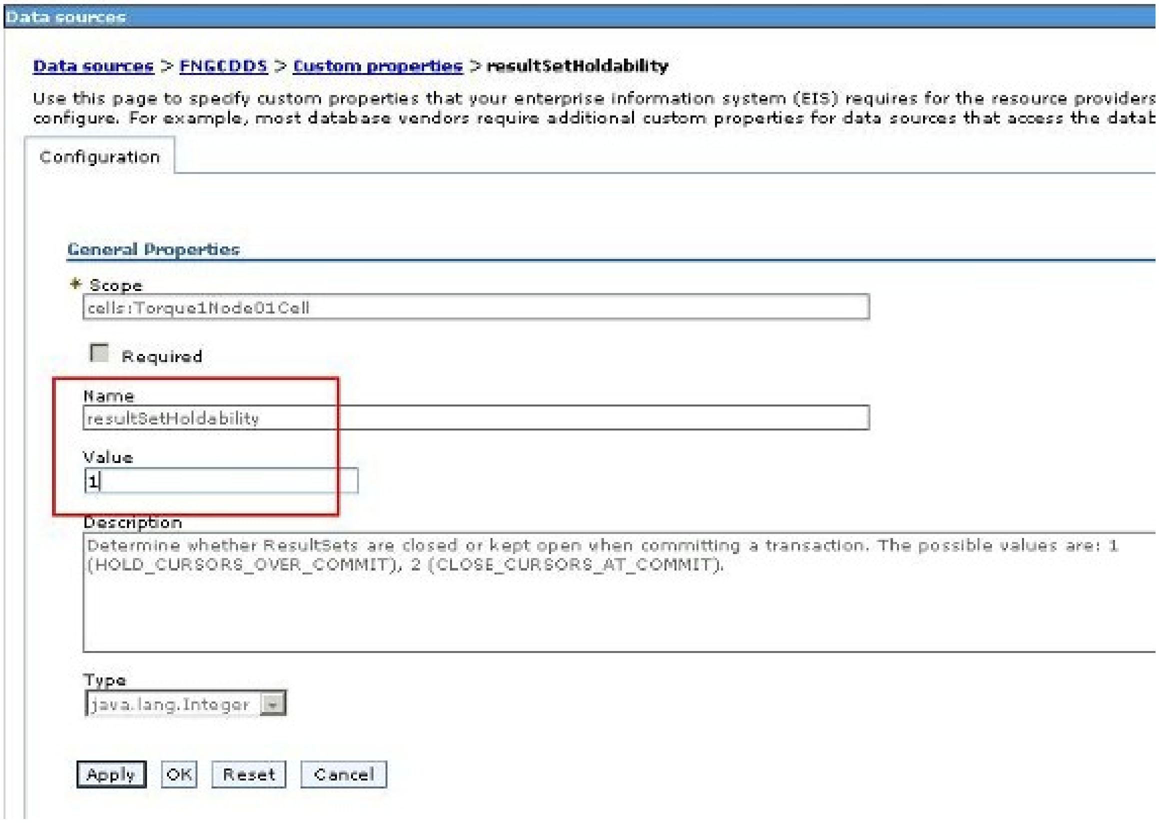
Additional Properties

- Connection pool properties
- WebSphere Application Server data :
- Custom properties

Related Items

- JAAS - J2C authentication data

Figure 24: JDBC non-XA data source Custom property “resultSetHoldability” – Set the value to 1



Data sources

Data sources > FNGCDDS > Custom properties > resultSetHoldability

Use this page to specify custom properties that your enterprise information system (EIS) requires for the resource providers configure. For example, most database vendors require additional custom properties for data sources that access the datab

General Properties

* Scope
cells:Torque1Node01Cell

□ Required

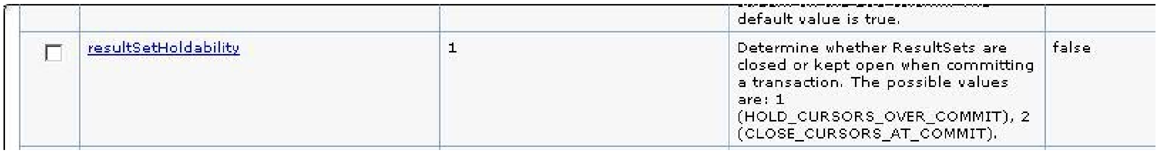
Name
resultSetHoldability

Value
1

Description
Determine whether ResultSets are closed or kept open when committing a transaction. The possible values are: 1 (HOLD_CURSORS_OVER_COMMIT), 2 (CLOSE_CURSORS_AT_COMMIT).

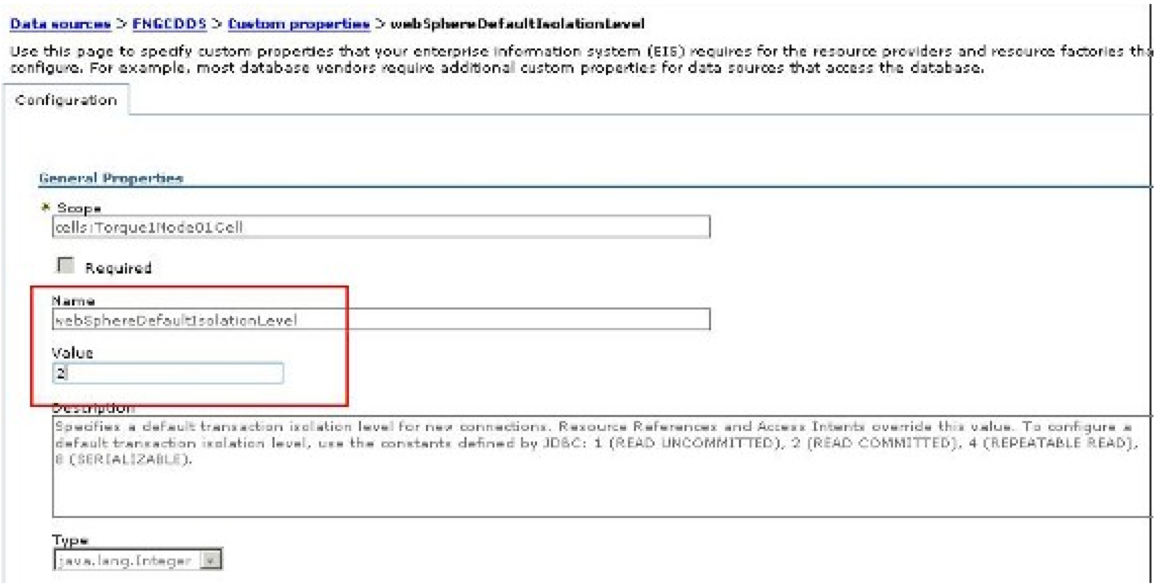
Type
java.lang.Integer

Figure 25: JDBC non-XA data source Custom property “resultSetHoldability” - afterwards



			default value is true.	
□	resultSetHoldability	1	Determine whether ResultSets are closed or kept open when committing a transaction. The possible values are: 1 (HOLD_CURSORS_OVER_COMMIT), 2 (CLOSE_CURSORS_AT_COMMIT).	false

Figure 26: JDBC non-XA data source Custom property “WebSphereDefaultIsolationLevel” - set to 2


Data sources > FNGCDDS > Custom properties > webSphereDefaultIsolationLevel

Use this page to specify custom properties that your enterprise information system (EIS) requires for the resource providers and resource factories th configure. For example, most database vendors require additional custom properties for data sources that access the database.

Configuration

General Properties

* Scope
cells:Torque1Node01Cell

□ Required

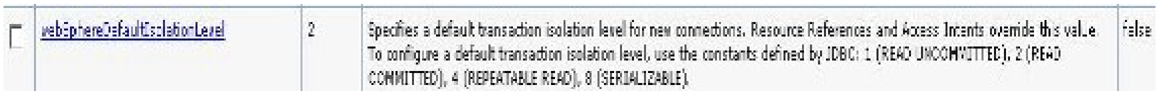
Name
webSphereDefaultIsolationLevel

Value
2

Description
Specifies a default transaction isolation level for new connections. Resource References and Access Intents override this value. To configure a default transaction isolation level, use the constants defined by JDBC: 1 (READ UNCOMMITTED), 2 (READ COMMITTED), 4 (REPEATABLE READ), 8 (SERIALIZABLE).

Type
java.lang.Integer

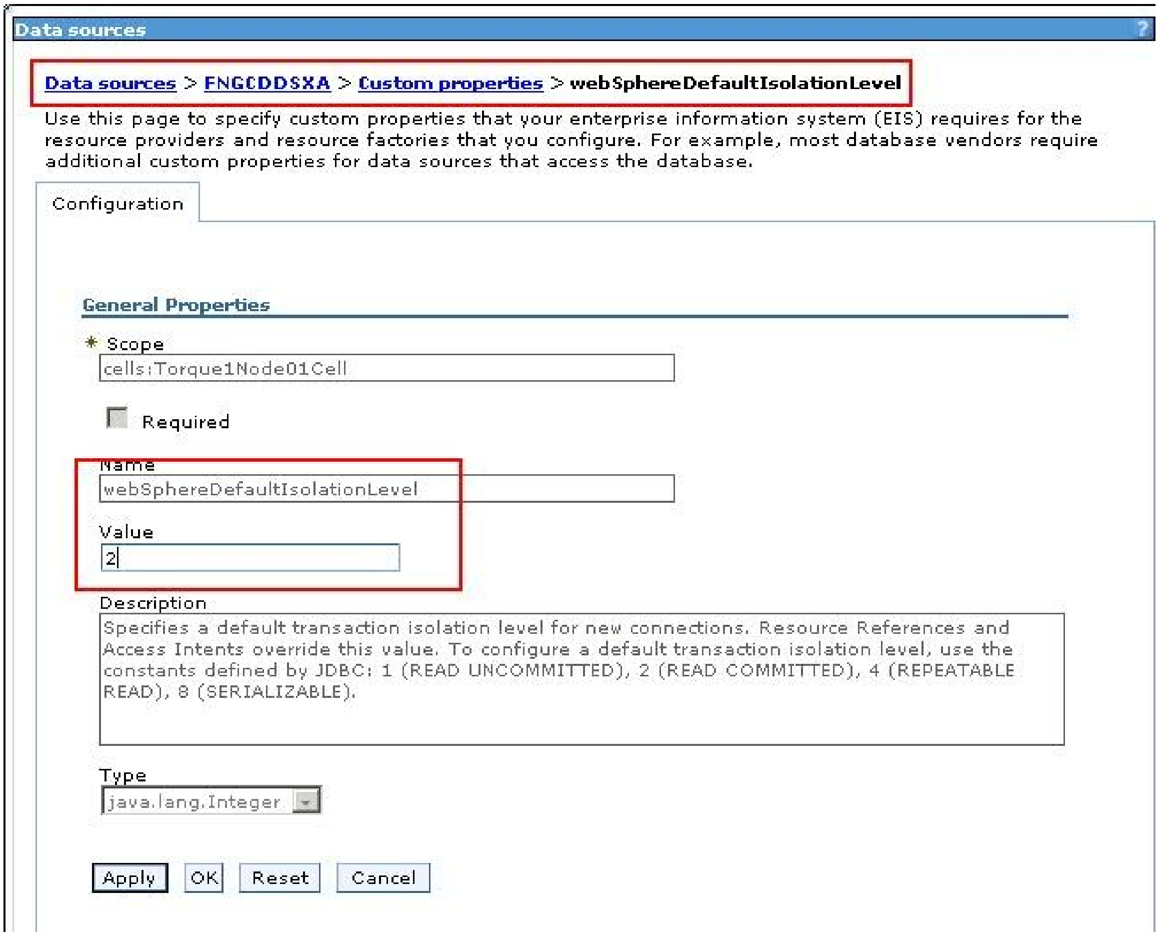
Figure 27: JDBC non-XA data source Custom property “WebSphereDefaultIsolationLevel” - afterwards



□	webSphereDefaultIsolationLevel	2	Specifies a default transaction isolation level for new connections. Resource References and Access Intents override this value. To configure a default transaction isolation level, use the constants defined by JDBC: 1 (READ UNCOMMITTED), 2 (READ COMMITTED), 4 (REPEATABLE READ), 8 (SERIALIZABLE).	false

For the JDBC XA data source, set the value of custom property "WebSphereDefaultIsolationLevel" to 2 as below:

Figure 28: JDBC XA data source Custom property "WebSphereDefaultIsolationLevel"



Data sources

Data sources > FNGCDDSXA > Custom properties > webSphereDefaultIsolationLevel

Use this page to specify custom properties that your enterprise information system (EIS) requires for the resource providers and resource factories that you configure. For example, most database vendors require additional custom properties for data sources that access the database.

Configuration

General Properties

* Scope
cells:Torque1Node01Cell

□ Required

Name
webSphereDefaultIsolationLevel

Value
2

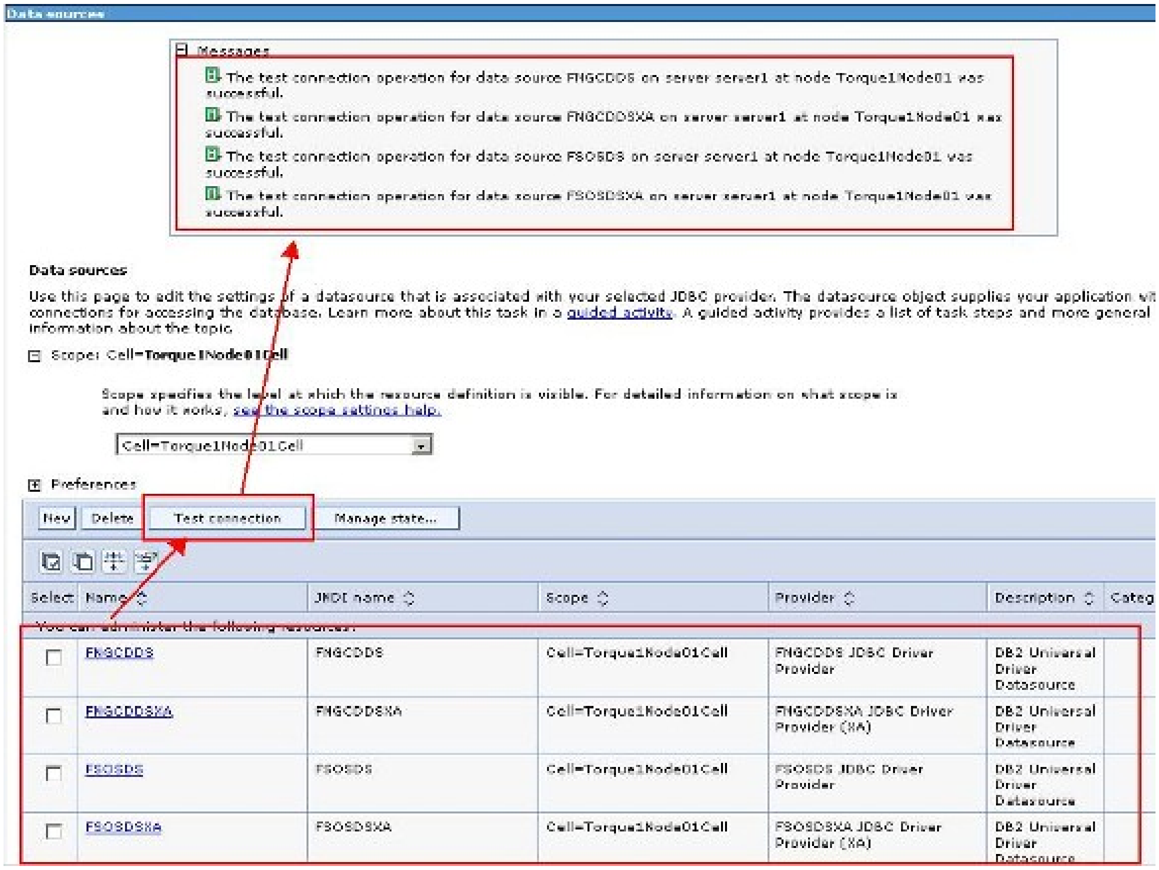
Description
Specifies a default transaction isolation level for new connections. Resource References and Access Intents override this value. To configure a default transaction isolation level, use the constants defined by JDBC: 1 (READ UNCOMMITTED), 2 (READ COMMITTED), 4 (REPEATABLE READ), 8 (SERIALIZABLE).

Type
java.lang.Integer

Similarly create the data sources (both XA and non-XA) for the object store database.

Once all the data sources and their properties are set, test the data sources to see if the connections are successful. If there is any error, it implies that there is some error talking to the databases. Tip: Make sure that the data base properties provided are correct

Figure 29: Testing the data source connections



Messages

The test connection operation for data source FNGCDDS on server server1 at node Torque1Node01 was successful.

The test connection operation for data source FNGCDDSXA on server server1 at node Torque1Node01 was successful.

The test connection operation for data source FSOSDS on server server1 at node Torque1Node01 was successful.

The test connection operation for data source FSOSDSXA on server server1 at node Torque1Node01 was successful.

Data sources

Use this page to edit the settings of a datasource that is associated with your selected JDBC provider. The datasource object supplies your application wi connections for accessing the database. Learn more about this task in a guided activity. A guided activity provides a list of task steps and more general information about the topic.

Scope: Cell=Torque1Node01Cell

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, see the scope settings help.

Cell=Torque1Node01Cell

You can administer the following resources:

Select	Name	JNDI name	Scope	Provider	Description	Categ
□	FNGCDDS	FNGCDDS	Cell=Torque1Node01Cell	FNGCDDS JDBC Driver Provider	DB2 Universal Driver Datasource	
□	FNGCDDSXA	FNGCDDSXA	Cell=Torque1Node01Cell	FNGCDDSXA JDBC Driver Provider (XA)	DB2 Universal Driver Datasource	
□	FSOSDS	FSOSDS	Cell=Torque1Node01Cell	FSOSDS JDBC Driver Provider	DB2 Universal Driver Datasource	
□	FSOSDSXA	FSOSDSXA	Cell=Torque1Node01Cell	FSOSDSXA JDBC Driver Provider (XA)	DB2 Universal Driver Datasource	

We, now, have successfully created the datasource objects that take care of talking to the underlying databases.

2. Configuring security (Login modules and LDAP configuration)

The next step would be to configure the authentication of the users accessing the P8 platform. This is done in two phases as below:

2.1 Configuring Login Modules

Content Platform Engine uses Java Authentication and Authorization Service (JAAS) framework for authentication only (not for Authorization). JAAS provides a policy-based, pluggable framework for reliably and securely determining who is invoking a Java application. The design of JAAS as a pluggable framework is a key architectural benefit of J2EE. It allows applications to remain independent of the underlying authentication technologies, and it allows third-party application server vendors, authentication providers, and single sign-on providers to package solutions that can be leveraged by all J2EE applications and clients. In an end user environment, new or updated single sign-on solutions can be plugged in without requiring modifications to the client and server applications that are already deployed.

The Content Platform Engine Enterprise Java Bean (EJB) resides within the J2EE Application Server's EJB container, and is therefore accessible only by authenticated callers, who can pass any authorization checks that the administrator has placed on the EJB.

To make use of a J2EE-based application, a client must first perform a JAAS login. To do this, the client must specify a JAAS configuration (typically through a configuration file). The JAAS configuration specifies the authentication technologies (LoginModules) that will be used to verify the client's credentials. A JAAS configuration file lists one entry for each configured application. Within an application's entry in a JAAS configuration file is a list of LoginModules for that application. When a client application attempts to authenticate, the JAAS framework dynamically determines the set of authentication technologies to invoke based on the contents of the configuration file.

Each entry in a JAAS configuration is marked as either Required, Requisite, Sufficient, or Optional. The authentication process for the client succeeds only if all LoginModules marked either Required or Requisite succeed. If no Required or Requisite LoginModules succeed, then at least one Sufficient or Optional LoginModule must succeed.

Content Platform Engine installation provides sample jaas login configuration file for different types of application servers, say, "jaas.conf.WebSphere" for WebSphere application server.

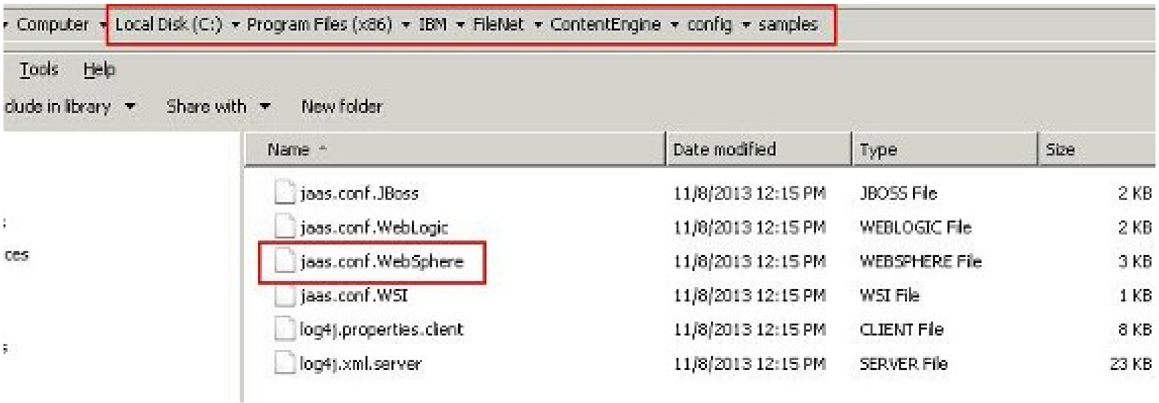
Listing 2. Sample JAAS configuration file

```
FileNetP8 {
    com.ibm.ws.security.common.auth.module.WSLoginModuleImpl required;
};

FileNetP8WSI {
    com.filenet.api.util.WSILoginModule required;
};


FileNetP8Engine {
    com.ibm.ws.security.common.auth.module.proxy.WSLoginModuleProxy required
        delegate=com.ibm.ws.security.common.auth.module.WSLoginModuleImpl;
};

FileNetP8Server {
    com.ibm.ws.security.common.auth.module.proxy.WSLoginModuleProxy required
        delegate=com.ibm.ws.security.common.auth.module.WSLoginModuleImpl;
};

FileNetP8KerberosService {
    com.ibm.ws.security.common.auth.module.proxy.WSLoginModuleProxy required
        delegate=com.filenet.engine.authentication.kerberos.login.KrbServiceLoginModule;
    com.ibm.ws.security.common.auth.module.proxy.WSLoginModuleProxy required
        delegate=com.ibm.ws.security.server.lm.ltpaLoginModule;
    com.ibm.ws.security.common.auth.module.proxy.WSLoginModuleProxy required
        delegate=com.ibm.ws.security.server.lm.wsMapDefaultInboundLoginModule;
};
```

```
Credentials {
    com.idoox.security.jaas.GSSLoginModule required debug=true;
};

ReceivedCredentials {
    com.idoox.security.jaas.GSSLoginModuleNoAuth required debug=true;
};

HttpCredentials {
    com.idoox.security.jaas.HttpLoginModule required debug=true;
};

KrbCredentials {
    com.ibm.security.auth.module.Krb5LoginModule required debug=false credsType=both;
};

CachedKrbCredentials {
    com.ibm.security.auth.module.Krb5LoginModule required debug=false
useDefaultCcache=true;
};

NamePasswordNoAN {
    com.idoox.security.jaas.NamePasswordLoginModuleNoAuth required debug=true;
};

NamePasswordAN {
    com.idoox.security.jaas.NamePasswordLoginModule required debug=true;
};

NameDigestAN {
    com.idoox.security.jaas.NameDigestLoginModule required debug=true;
};

NameMapping {
    com.idoox.security.jaas.NameLoginModuleNoAuth required debug=true;
};

CertsMapping {
    com.idoox.security.jaas.CertsLoginModule required debug=true;
};

Navigator {
    com.ibm.ws.security.common.auth.module.proxy.WSLoginModuleProxy required
        delegate=com.ibm.ws.security.server.lm.ltpaLoginModule;
    com.ibm.ws.security.common.auth.module.proxy.WSLoginModuleProxy required
        delegate=com.ibm.ws.security.server.lm.wsMapDefaultInboundLoginModule;
};
```

The sample JAAS configuration provided by FileNet Content Platform Engine installation are located at <CPE_InstallFolder>\config\samples

Figure 30: Location of sample JAAS configuration



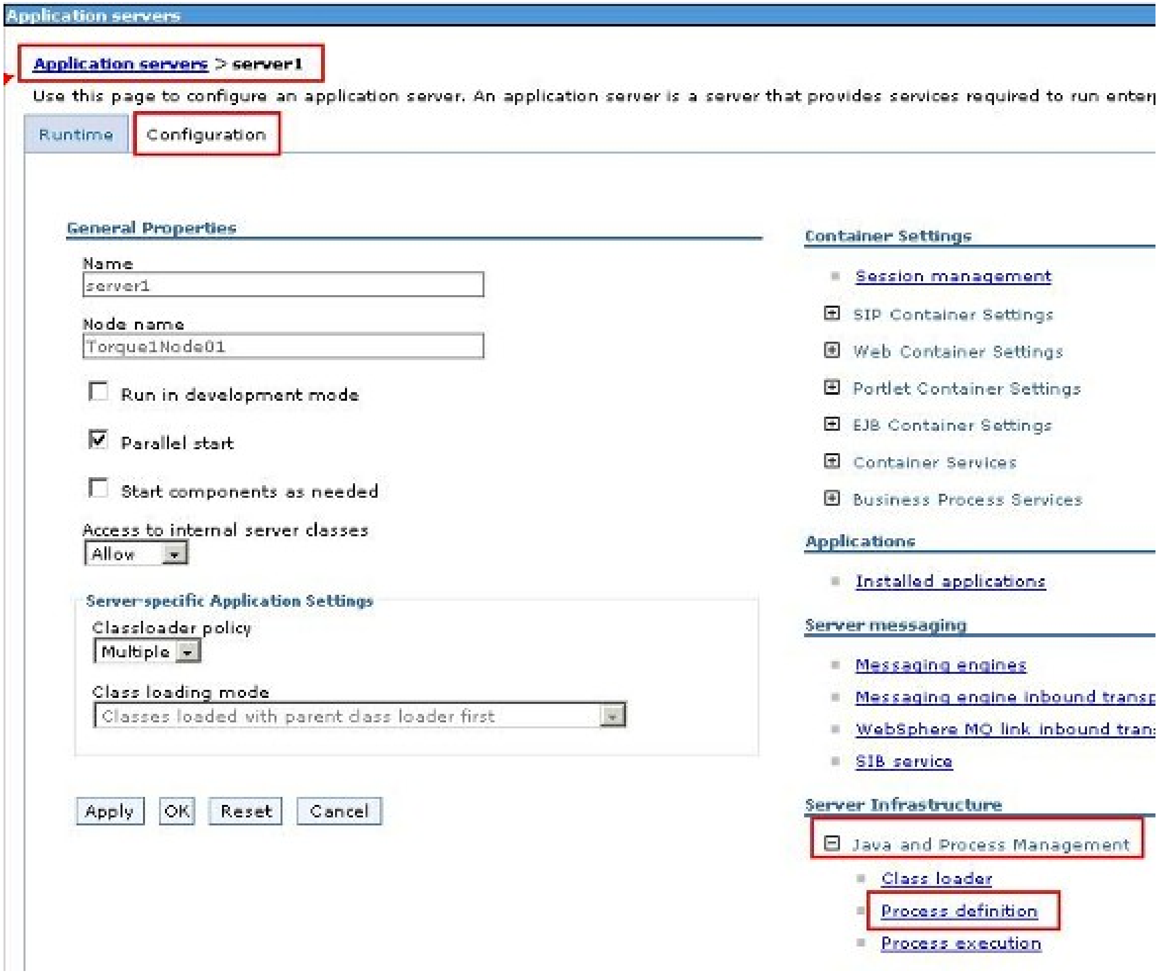
Computer ▾ Local Disk (C:) ▾ Program Files (x86) ▾ IBM ▾ FileNet ▾ ContentEngine ▾ config ▾ samples
Tools
Help
clude in library ▾
Share with ▾
New folder
ces

Name	Date modified	Type	Size
jaas.conf.JBoss	11/8/2013 12:15 PM	JBOSS File	2 KB
jaas.conf.WebLogic	11/8/2013 12:15 PM	WEBLOGIC File	2 KB
jaas.conf.WebSphere	11/8/2013 12:15 PM	WEBSPHERE File	3 KB
jaas.conf.WSI	11/8/2013 12:15 PM	WSI File	1 KB
log4j.properties.client	11/8/2013 12:15 PM	CLIENT File	8 KB
log4j.xml.server	11/8/2013 12:15 PM	SERVER File	23 KB

The WebSphere JAAS configuration file can be specified using the JVM argument "-Djava.security.auth.login.conf".

In WebSphere admin console, navigate to server > "Java and Process Management" > "Process Definition"

Figure 31: Process Definition entry for the server instance



Application servers

Application servers > server1

Use this page to configure an application server. An application server is a server that provides services required to run enter

General Properties

Name: server1

Node name: Torque1Node01

□ Run in development mode

☑ Parallel start

□ Start components as needed

Access to internal server classes: Allow

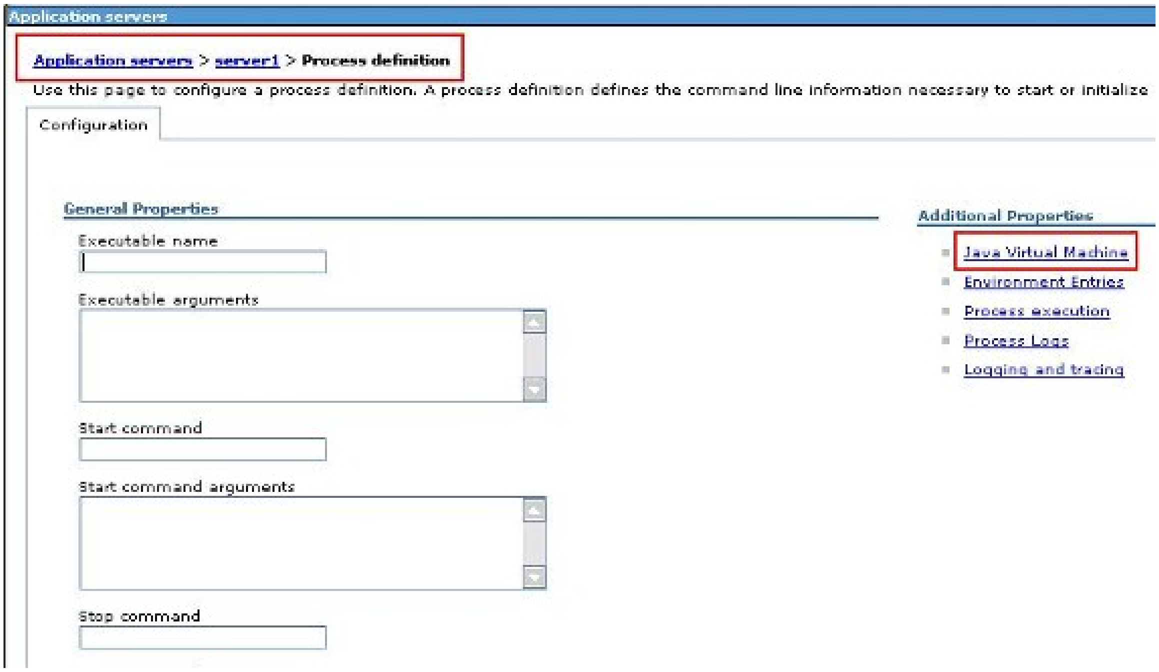
Server-specific Application Settings

Classloader policy: Multiple

Class loading mode: Classes loaded with parent class loader first

Navigate to the JVM properties

Figure 32: Java Virtual Machine entry for the server instance



Application servers

Application servers > server1 > Process definition

Use this page to configure a process definition. A process definition defines the command line information necessary to start or initialize

Configuration

General Properties

Executable name

Executable arguments

Start command

Start command arguments

Stop command

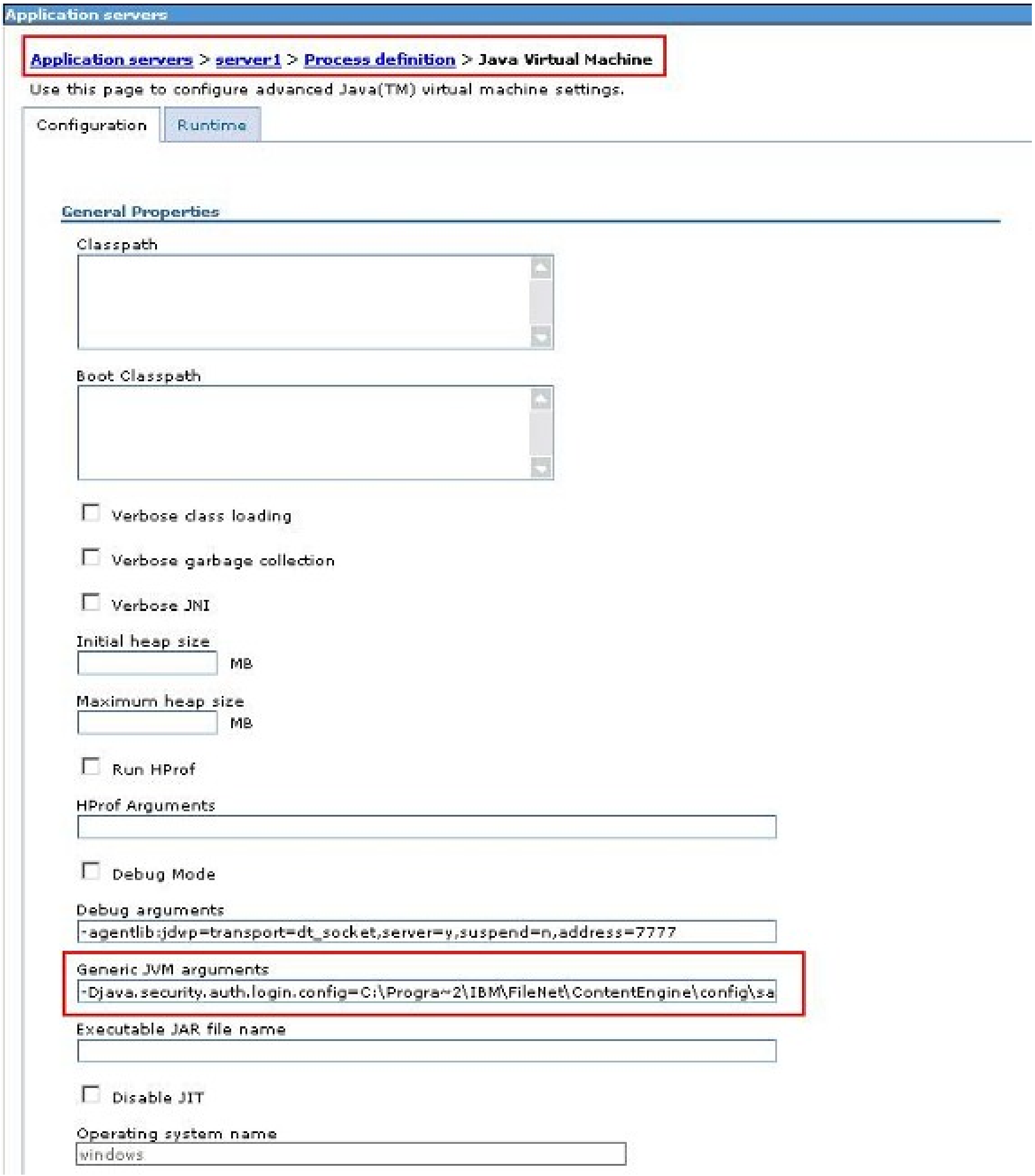
Additional Properties

- Java Virtual Machine
- Environment Entries
- Process execution
- Process Logs
- Logging and tracing

Set the Generic JVM argument as below:

```
-Djava.security.auth.login.config=<CPE_InstallFolder>\config\samples\jaas.conf.WebSphere
```

Figure 33: Setting the JVM argument for login module configuration



Application servers > server1 > Process definition > Java Virtual Machine

Use this page to configure advanced Java(TM) virtual machine settings.

General Properties

Classpath

Boot Classpath

□ Verbose class loading

□ Verbose garbage collection

□ Verbose JNI

Initial heap size
MB

Maximum heap size
MB

□ Run HProf

HProf Arguments

□ Debug Mode

Debug arguments
-agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=7777

Generic JVM arguments
-Djava.security.auth.login.config=C:\Progra~2\IBM\FileNet\ContentEngine\config\sa

Executable JAR file name

□ Disable JIT

Operating system name
windows

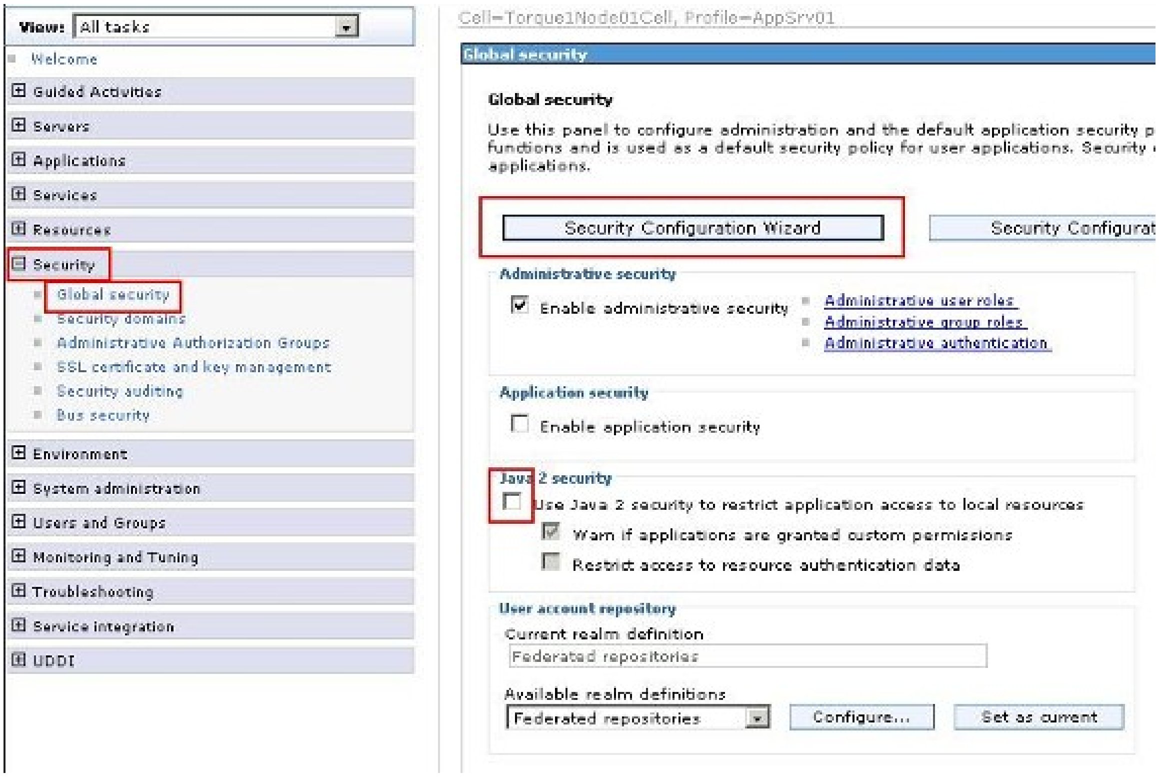
2.2 Configuring LDAP

IBM FileNet P8 platform doesn't maintain the list of users and groups needed to access the Content Platform Engine. Instead, it relies on the LDAP directory configured by the user in the application server. The LDAP directory is solely responsible for managing (i.e. add/update/delete) the users or groups that are required for access to Content Platform Engine.

Content Platform Engine retrieves security data from directory servers for the purpose of authenticating security principals and authorizing users and groups. To retrieve this data, an LDAP account should be granted as the WebSphere Application Server administrative role. The following steps describe how an LDAP directory is configured in Application Server:

Run the “security configuration wizard” from Security > Global Security > Start Configuration Wizard

Figure 34: Security Configuration Wizard



Cell=Torque1Node01Cell, Profile=AppSrv01

Global security

Use this panel to configure administration and the default application security p functions and is used as a default security policy for user applications. Security (applications.

Administrative security

☑ Enable administrative security

Application security

□ Enable application security

Java 2 security

□ Use Java 2 security to restrict application access to local resources

☑ Warn if applications are granted custom permissions

□ Restrict access to resource authentication data

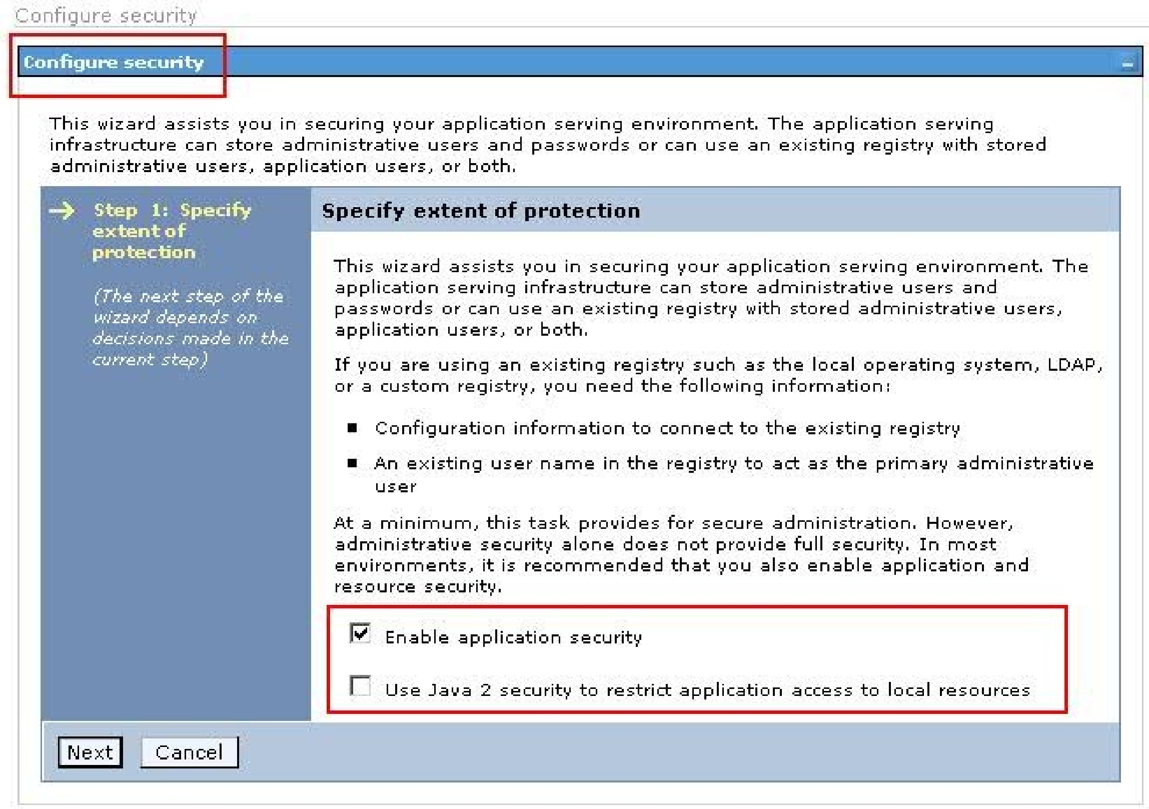
User account repository

Current realm definition: Federated repositories

Available realm definitions: Federated repositories

IBM IBM FileNet P8 platform utilizes LDAP-based security, and does not support Java 2 security.

Figure 35: Configuring Security



Configure security

Configure security

This wizard assists you in securing your application serving environment. The application serving infrastructure can store administrative users and passwords or can use an existing registry with stored administrative users, application users, or both.

→ **Step 1: Specify extent of protection**

(The next step of the wizard depends on decisions made in the current step)

Specify extent of protection

This wizard assists you in securing your application serving environment. The application serving infrastructure can store administrative users and passwords or can use an existing registry with stored administrative users, application users, or both.

If you are using an existing registry such as the local operating system, LDAP, or a custom registry, you need the following information:

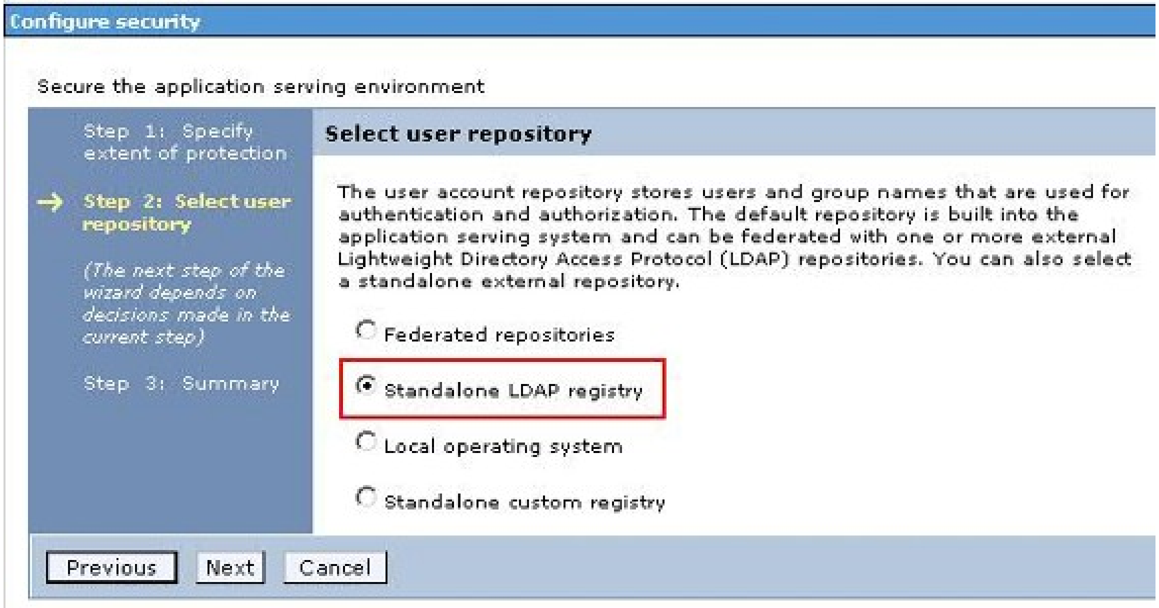
- Configuration information to connect to the existing registry
- An existing user name in the registry to act as the primary administrative user

At a minimum, this task provides for secure administration. However, administrative security alone does not provide full security. In most environments, it is recommended that you also enable application and resource security.

☑ Enable application security

□ Use Java 2 security to restrict application access to local resources

Figure 36: Select user repository



Configure security

Secure the application serving environment

Step 1: Specify extent of protection

→ **Step 2: Select user repository**

(The next step of the wizard depends on decisions made in the current step)

Step 3: Summary

Select user repository

The user account repository stores users and group names that are used for authentication and authorization. The default repository is built into the application serving system and can be federated with one or more external Lightweight Directory Access Protocol (LDAP) repositories. You can also select a standalone external repository.

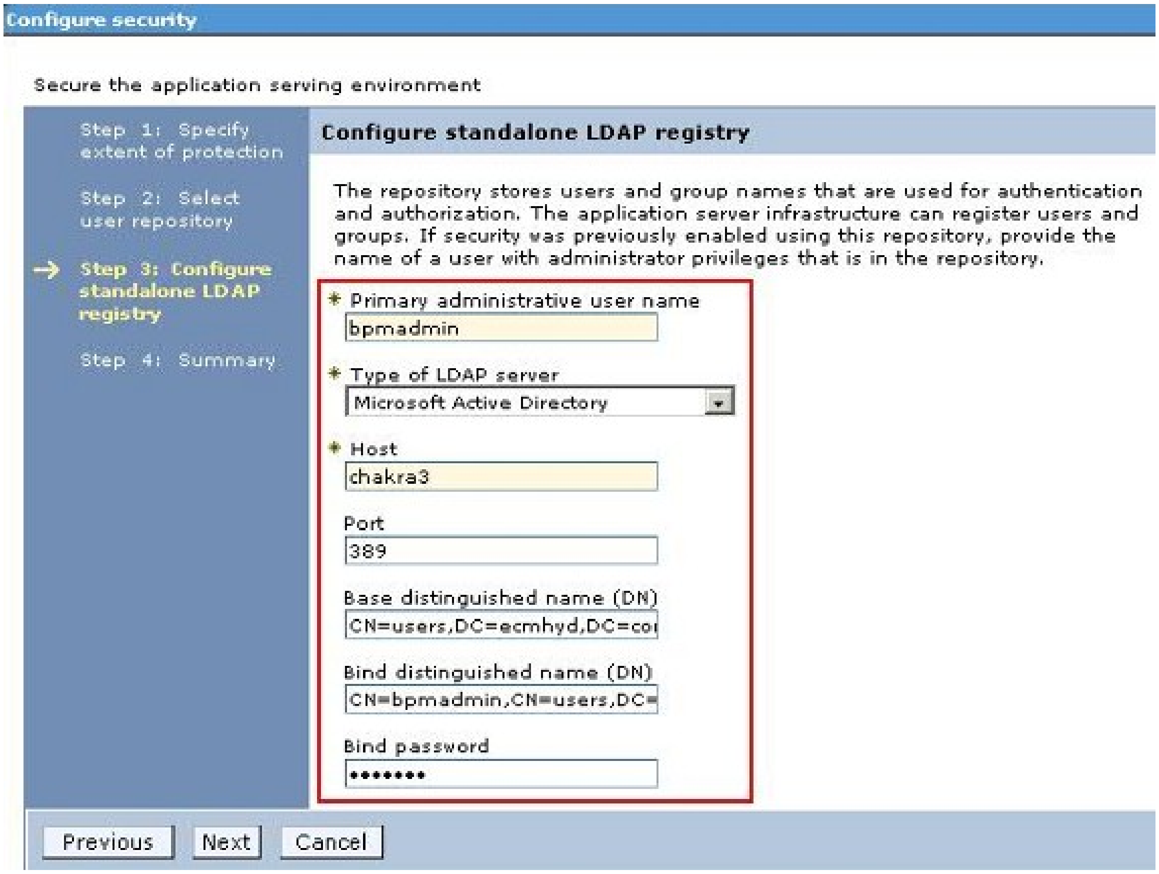
○ Federated repositories

● Standalone LDAP registry

○ Local operating system

○ Standalone custom registry

Figure 37: Configure Standalone LDAP registry



Configure security

Secure the application serving environment

Step 1: Specify extent of protection
Step 2: Select user repository
→ Step 3: Configure standalone LDAP registry
Step 4: Summary

Configure standalone LDAP registry

The repository stores users and group names that are used for authentication and authorization. The application server infrastructure can register users and groups. If security was previously enabled using this repository, provide the name of a user with administrator privileges that is in the repository.

* Primary administrative user name: bpmadmin

* Type of LDAP server: Microsoft Active Directory

* Host: chakra3

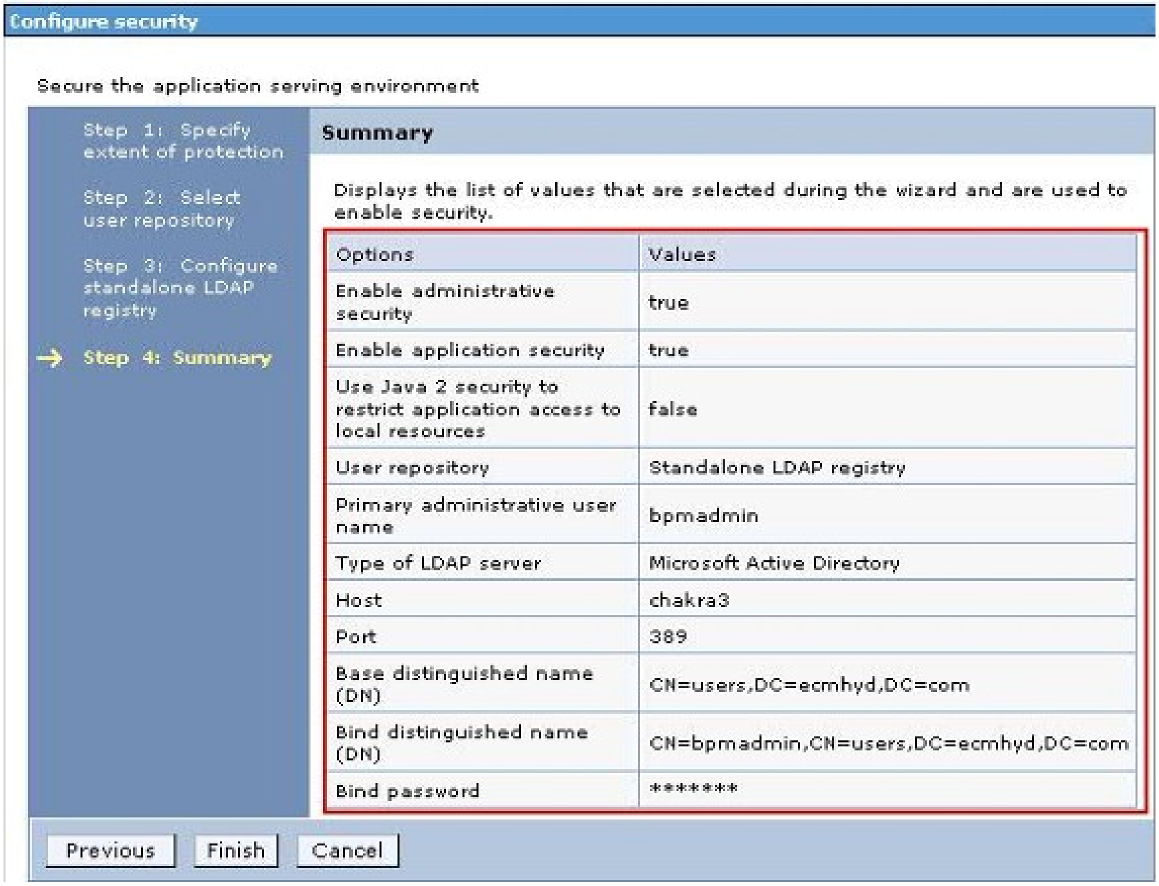
Port: 389

Base distinguished name (DN): CN=users,DC=ecmhyd,DC=coı

Bind distinguished name (DN): CN=bpmadmin,CN=users,DC=

Bind password: ••••••••

Figure 38: Summary



Secure the application serving environment

Step 1: Specify extent of protection
Step 2: Select user repository
Step 3: Configure standalone LDAP registry
→ Step 4: Summary

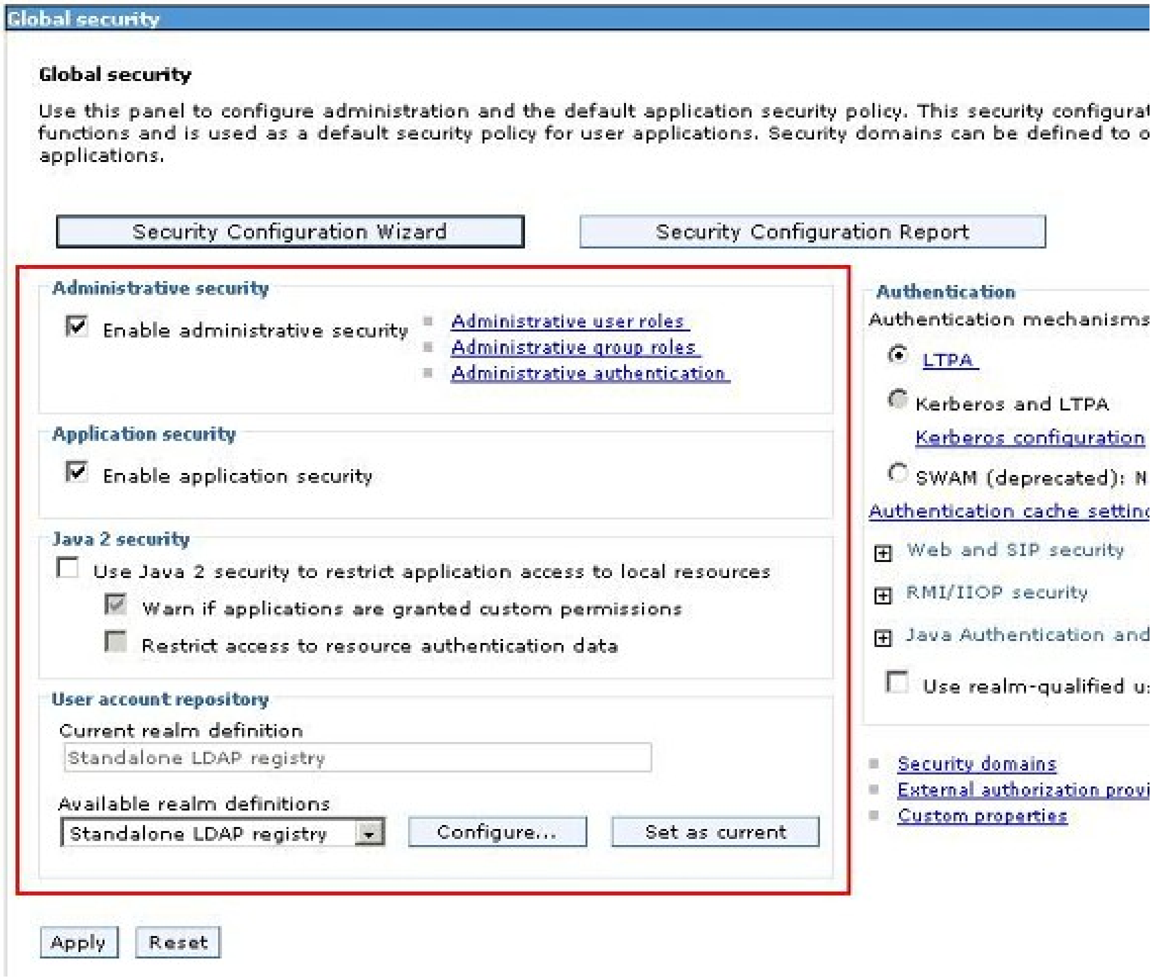
Summary

Displays the list of values that are selected during the wizard and are used to enable security.

Options	Values
Enable administrative security	true
Enable application security	true
Use Java 2 security to restrict application access to local resources	false
User repository	Standalone LDAP registry
Primary administrative user name	bpmadmin
Type of LDAP server	Microsoft Active Directory
Host	chakra3
Port	389
Base distinguished name (DN)	CN=users,DC=ecmhyd,DC=com
Bind distinguished name (DN)	CN=bpmadmin,CN=users,DC=ecmhyd,DC=com
Bind password	*******

Once created, make sure that Application security is selected and Java 2 security is not selected.

Figure 39: Global Security



Global security

Global security

Use this panel to configure administration and the default application security policy. This security configurat functions and is used as a default security policy for user applications. Security domains can be defined to o applications.

Security Configuration Wizard | Security Configuration Report

Administrative security

☑ Enable administrative security
- Administrative user roles
- Administrative group roles
- Administrative authentication

Application security

☑ Enable application security

Java 2 security

☐ Use Java 2 security to restrict application access to local resources

☑ Warn if applications are granted custom permissions

☐ Restrict access to resource authentication data

User account repository

Current realm definition: Standalone LDAP registry

Available realm definitions: Standalone LDAP registry ▾ | Configure... | Set as current

Authentication

Authentication mechanisms

● LTPA

○ Kerberos and LTPA

Kerberos configuration

○ SWAM (deprecated): N

Authentication cache settinc

⊞ Web and SIP security

⊞ RMI/IIOP security

⊞ Java Authentication and

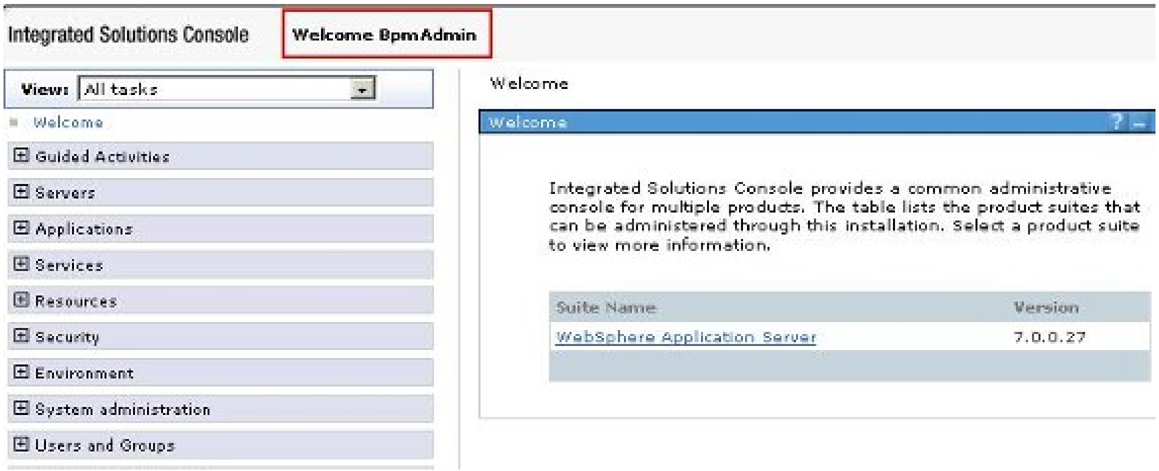
☐ Use realm-qualified u:

- Security domains
- External authorization provi
- Custom properties

Apply | Reset

After the Global Security is configured in application server, the application server needs to be restarted. After the restart, only the users from the LDAP server who are configured as administrator would be able to login into WebSphere admin console.

Figure 40: Login into WebSphere using LDAP user configured as administrator



Integrated Solutions Console provides a common administrative console for multiple products. The table lists the product suites that can be administered through this installation. Select a product suite to view more information.

Suite Name	Version
WebSphere Application Server	7.0.0.27

This completes the Configuration of Authentication for Content Platform Engine.

2.3 Authorization

When a security principal that has already been authenticated attempts to access FileNet P8 objects, Content Platform Engine will attempt to retrieve that principal's user and group memberships from the directory service provider. If successful, the user or group will be authorized to carry out actions described by the access rights placed on the objects. This Authorization is defined on the respective object that resides inside the P8 domain with the help of administration tools like Administration Console for Content Platform Engine (ACCE) and FileNet Enterprise Manager (FEM). (Authorization, in detail, is not going to be covered in the scope of this article)

3. Bootstrapping the Content Platform Engine EAR

Content Platform Engine needs a directory service account to establish a connection with the application server, access the application server's JNDI tree, look up the data sources for accessing the GCD, and start up Content Platform Engine's background tasks. This directory service account is called the “bootstrap user” and it is needed to create the Global Configuration Database (GCD), and thereafter to provide the resources it needs to boot up. Content Platform Engine will not be able to start if this user is not able to authenticate.

This “bootstrap user” information is configured inside the Content Platform Engine EAR file under the file named as “CEMPBoot.properties”. Once Content Platform Engine is

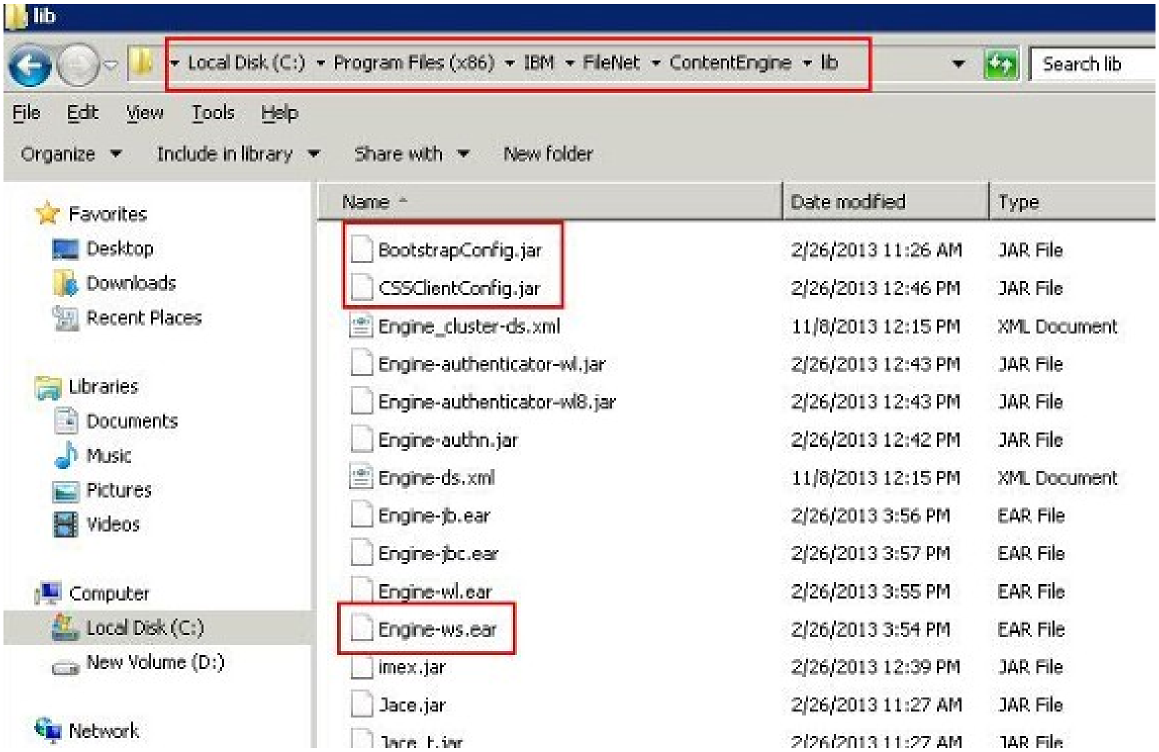
configured and the new P8 domain is created and functioning, the bootstrap file continues to provide the information to allow Content Platform Engine to load.
Bootstrapping of CPE EAR file with the directory service account is performed using the following steps:

3.1 Creating Content Platform Engine Bootstrap account

Content Platform Engine installation provides the utility “BootstrapConfig.jar” that can be used to add/modify the bootstrap information into the CPE EAR file Engine-xx.ear (which is later deployed into Application Server).
Note: CPE installation provides different EAR files for different application servers, which can be identified based on the xx in “Engine-xx.ear” where “ws” refers to WebSphere, “wl” refers to WebLogic, “jb” refers to JBoss, “jbc” refers to JBoss Cluster.

This utility gets installed into “<CPE_InstallFolder>\lib” directory. Also, contained in this directory, is the Content Platform Engine's EAR file and another configuration utility for Text Extraction (explained later).

Figure 41: Location of Bootstrap and Text Extraction utilities

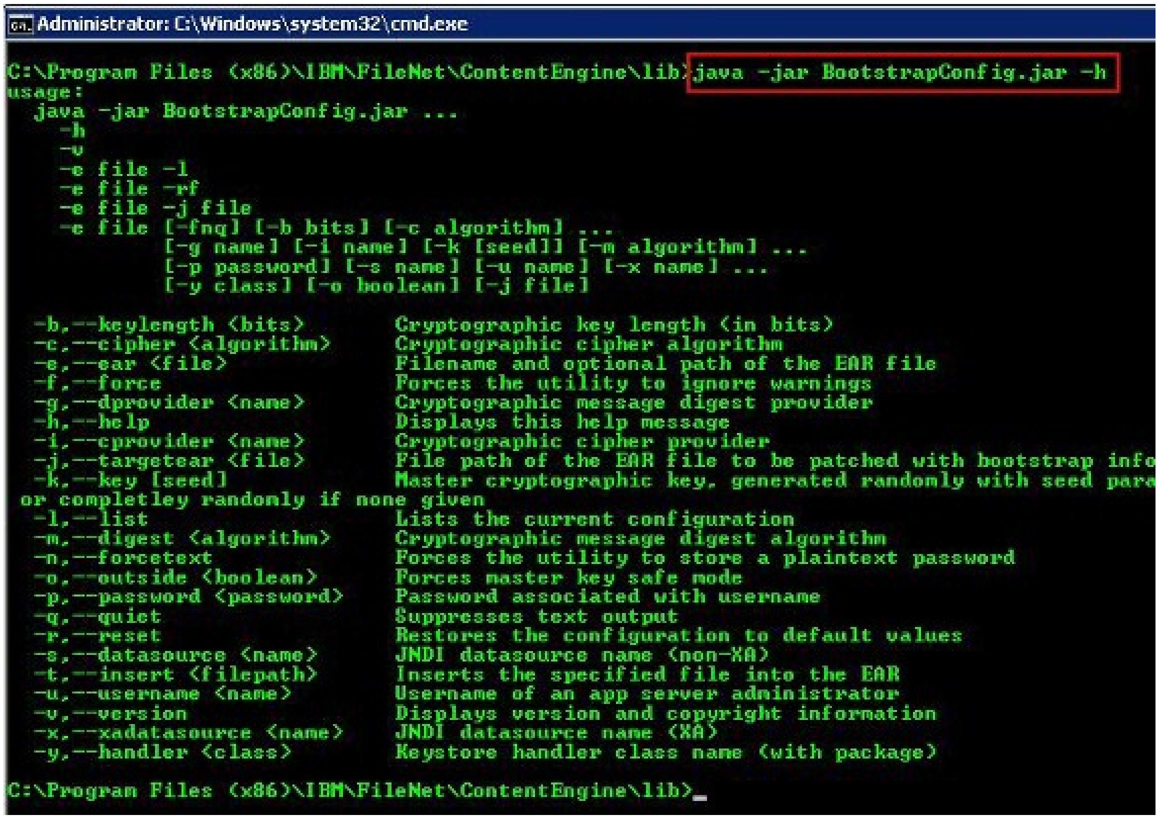


Name	Date modified	Type
BootstrapConfig.jar	2/26/2013 11:26 AM	JAR File
CSSClientConfig.jar	2/26/2013 12:46 PM	JAR File
Engine_cluster-ds.xml	11/8/2013 12:15 PM	XML Document
Engine-authenticator-wl.jar	2/26/2013 12:43 PM	JAR File
Engine-authenticator-wl8.jar	2/26/2013 12:43 PM	JAR File
Engine-authn.jar	2/26/2013 12:42 PM	JAR File
Engine-ds.xml	11/8/2013 12:15 PM	XML Document
Engine-jb.ear	2/26/2013 3:56 PM	EAR File
Engine-jbc.ear	2/26/2013 3:57 PM	EAR File
Engine-wl.ear	2/26/2013 3:55 PM	EAR File
Engine-ws.ear	2/26/2013 3:54 PM	EAR File
imex.jar	2/26/2013 12:39 PM	JAR File
Jace.jar	2/26/2013 11:27 AM	JAR File
Jace_t.jar	2/26/2013 11:27 AM	JAR File

Open command prompt and navigate to the Content Platform Engine's lib folder.
Run the command “java -jar BootstrapConfig.jar -h” to know all the options that are provided by this utility. (Note: The list of options might increase in higher releases.)

This bootstrap utility can be used either in a fresh installation where it adds the details of the user, the GCD data sources, SHA encrypted password information into the new EAR (OR) to update an EAR file, usually in upgrade scenarios, with the information residing in an already deployed EAR file.

Figure 42: BootstrapConfig.jar options

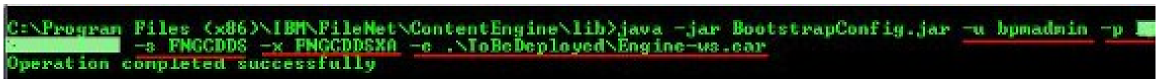


Administrator: C:\Windows\system32\cmd.exe

```
C:\Program Files (x86)\IBM\FileNet\ContentEngine\lib>java -jar BootstrapConfig.jar -h
usage:
  java -jar BootstrapConfig.jar ...
    -h
    -v
    -e file -l
    -e file -rf
    -e file -j file
    -e file [-fnq] [-b bits] [-c algorithm] ...
            [-g name] [-i name] [-k [seed]] [-m algorithm] ...
            [-p password] [-s name] [-u name] [-x name] ...
            [-y class] [-o boolean] [-j file]

  -b,--keylength <bits>        Cryptographic key length (in bits)
  -c,--cipher <algorithm>      Cryptographic cipher algorithm
  -e,--ear <file>              Filename and optional path of the EAR file
  -f,--force                   Forces the utility to ignore warnings
  -g,--dprovider <name>        Cryptographic message digest provider
  -h,--help                    Displays this help message
  -i,--cprovider <name>        Cryptographic cipher provider
  -j,--targetear <file>        File path of the EAR file to be patched with bootstrap info
  -k,--key [seed]              Master cryptographic key, generated randomly with seed para
 or completley randomly if none given
  -l,--list                    Lists the current configuration
  -m,--digest <algorithm>      Cryptographic message digest algorithm
  -n,--forcetext               Forces the utility to store a plaintext password
  -o,--outside <boolean>       Forces master key safe mode
  -p,--password <password>     Password associated with username
  -q,--quiet                   Suppresses text output
  -r,--reset                   Restores the configuration to default values
  -s,--datasource <name>       JNDI datasource name (non-XA)
  -t,--insert <filepath>       Inserts the specified file into the EAR
  -u,--username <name>         Username of an app server administrator
  -v,--version                 Displays version and copyright information
  -x,--xadatasource <name>     JNDI datasource name (XA)
  -y,--handler <class>         Keystore handler class name (with package)

C:\Program Files (x86)\IBM\FileNet\ContentEngine\lib>_
```

For a fresh installation, the information about the username, password, datasource name connecting to the GCD database, XA datasource name for the GCD database, and the location of the EAR file to be be bootstrapped should be specified as below:
Run the command `“java -jar BoostrapConfig.jar -u <username> -p <password> -s <GCD_DatasourceName> -x <GCD_XADatasourceName> -e <EARFiletoBeDeployed>”`

Figure 43: Bootstrapping an EAR for fresh installation

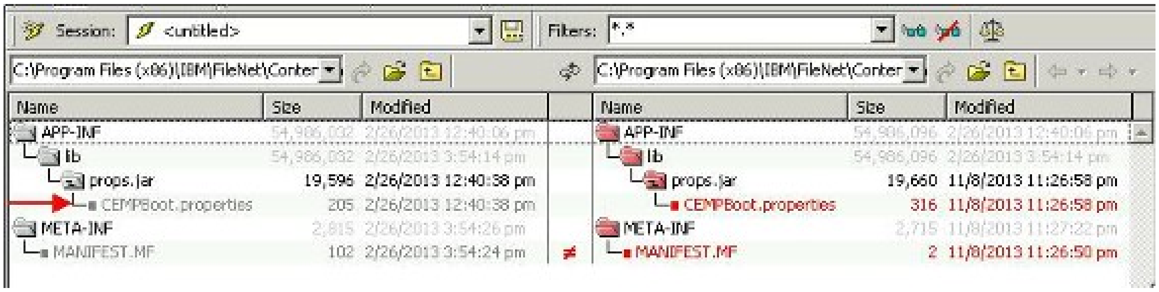


```
C:\Program Files (x86)\IBM\FileNet\ContentEngine\lib>java -jar BootstrapConfig.jar -u bpmadmin -p [redacted]
 -s FNGCDDS -x FNGCDDSXA -e .\ToBeDeployed\Engine-ws.ear
Operation completed successfully
```

Note: I have copied the newly installed EAR file into a sub-directory “ToBeDeployed” for precautionary measure. One can directory run the bootstrap command directly on the EAR file that gets installed into the “lib” folder.

Once, the EAR is bootstrapped, the information is stored in the file “CEMPBoot.properties” located under “APP-INF/lib/props.jar/” inside the CPE EAR file.

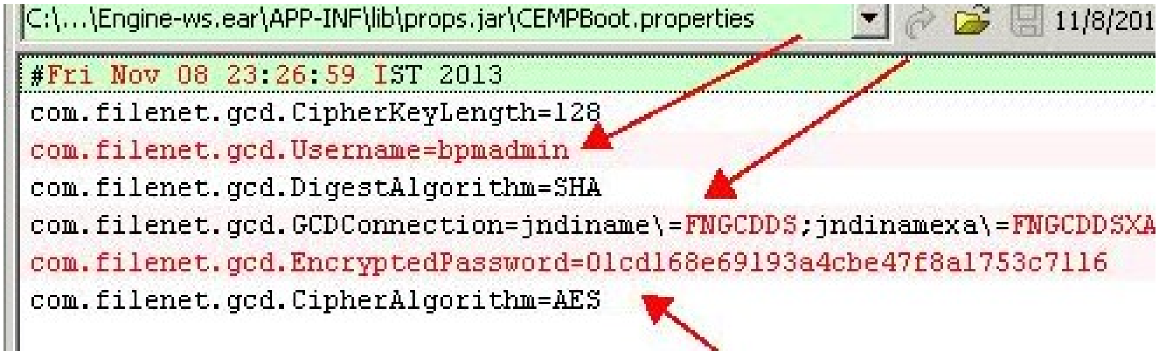
Figure 44: File changes in CPE EAR i.e. “Engine-xx.ear” file



Session: <untitled>
Filters: *.*
C:\Program Files (x86)\IBM\FileNet\Conter
C:\Program Files (x86)\IBM\FileNet\Conter

Name	Size	Modified		Name	Size	Modified
APP-INF	54,986,032	2/26/2013 12:40:06 pm		APP-INF	54,986,096	2/26/2013 12:40:06 pm
lib	54,986,032	2/26/2013 3:54:14 pm		lib	54,986,096	2/26/2013 3:54:14 pm
props.jar	19,596	2/26/2013 12:40:38 pm		props.jar	19,660	11/8/2013 11:26:58 pm
CEMPBoot.properties	205	2/26/2013 12:40:38 pm		CEMPBoot.properties	316	11/8/2013 11:26:58 pm
META-INF	2,815	2/26/2013 3:54:26 pm		META-INF	2,715	11/8/2013 11:27:22 pm
MANIFEST.MF	102	2/26/2013 3:54:24 pm	≠	MANIFEST.MF	2	11/8/2013 11:26:50 pm

Figure 45: Changes inside the CEMPBoot.properties



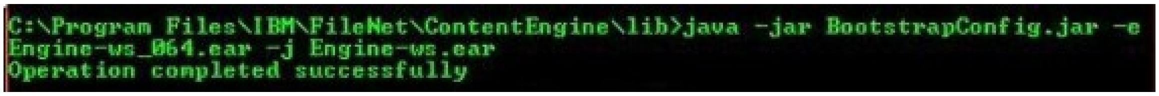
C:\...\Engine-ws.ear\APP-INF\lib\props.jar\CEMPBoot.properties
11/8/201

```
#Fri Nov 08 23:26:59 IST 2013
com.filenet.gcd.CipherKeyLength=128
com.filenet.gcd.Username=bpmadmin
com.filenet.gcd.DigestAlgorithm=SHA
com.filenet.gcd.GCDConnection=jndiname\=FNGCDDS;jndinamexa\=FNGCDDSXA
com.filenet.gcd.EncryptedPassword=01cd168e69193a4cbe47f8a1753c7116
com.filenet.gcd.CipherAlgorithm=AES
```

In case of an upgrade environment, the bootstrap utility takes the information from the old EAR file and places them into the new EAR file that gets laid with the installation.
To bootstrap the EAR for an upgraded environment, run the command

```
java -jar BootstrapConfig.jar -e <AlreadyDeployedEAR> - j <newEARToBeDeployed>
```

Figure 46: Bootstrapping CPE EAR for an Upgrade Configuration



```
C:\Program Files\IBM\FileNet\ContentEngine\lib>java -jar BootstrapConfig.jar -e
Engine-ws_064.ear -j Engine-ws.ear
Operation completed successfully
```

Note: The figure represents a sample configuration where the old EAR file “Engine-ws_064.ear” (renamed to distinguish, only for internal understanding) is used.

3.2 Text Extraction Utility

Starting CPE 5.2.0, a CSS Server Client file (Text Extraction utility) is placed inside the CPE EAR file to be deployed. This CSS server client is supported only on 64-bit Operating systems and hence, CPE is required to be installed on a 64-bit machine. So, the next step in configuration would be to add the CSS server client tool to the already bootstrapped CPE EAR file.

CPE Installation provides a CSS Client configuration tool “CSSClientTool” in the form of “CSSClientConfig.jar” (located at the “<CPE_InstallFolder>\lib” directory) to embed the CSS Server Client file into the CPE EAR file. Running this utility needs “commons-cli-1.0.jar” to reside in the class path. This is a third party library (Apache Commons CLI library) that provides an API for parsing command line options passed to programs. This “commons-cli-1.0.jar” can be found inside the “org.apache.commons.cli_1.0.0.jar” file that is laid for FileNet Deployment Manager tool during the Content Platform Engine installation and is located at “<CPE_InstallFolder>\tools\deploy\plugins”. Extract the jar file using any unzip utility.

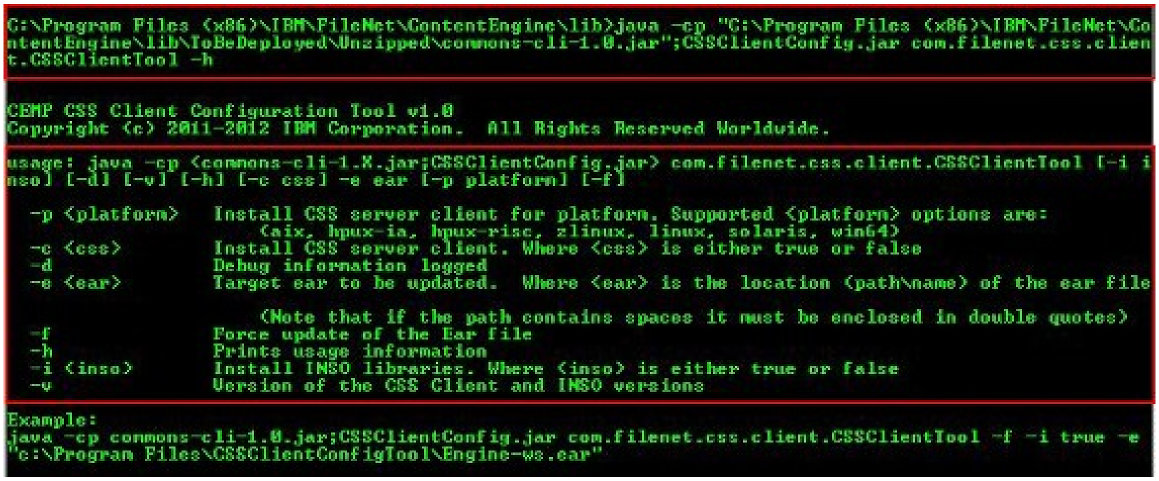
Following steps would add this CSS Server client tool to the already bootstrapped EAR file.

In command prompt, navigate to the lib folder in the <CPE_InstallFolder> location and run the CSSClientTool using the command

```
java  -cp  <commons-cli-1.0.jar;CSSClientConfig.jar>  com.filenet.css.client.CSSClientTool
-d -v -c false -e <PathToBootstrappedEAR> -p <platform> -f
```

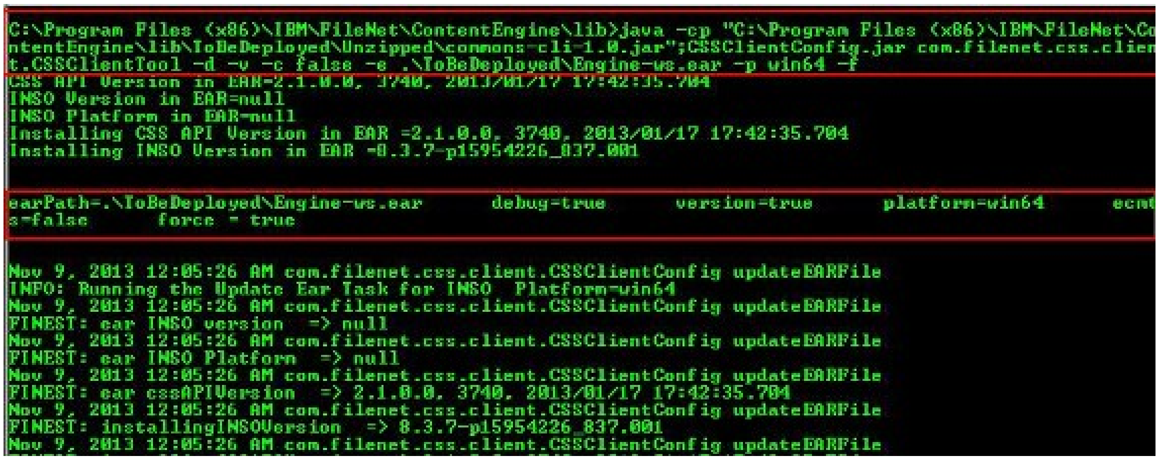
The descriptions of the options can be seen in the following figure.. (“-d” for logging debug information, “-v” for printing the version of CSS Client and INSO versions, “-c false” to specify not to install the CSS Client, “-p win64” for installing on a 64-bit windows platform, “-f” for force update)

Figure 47: CSSClientTool options



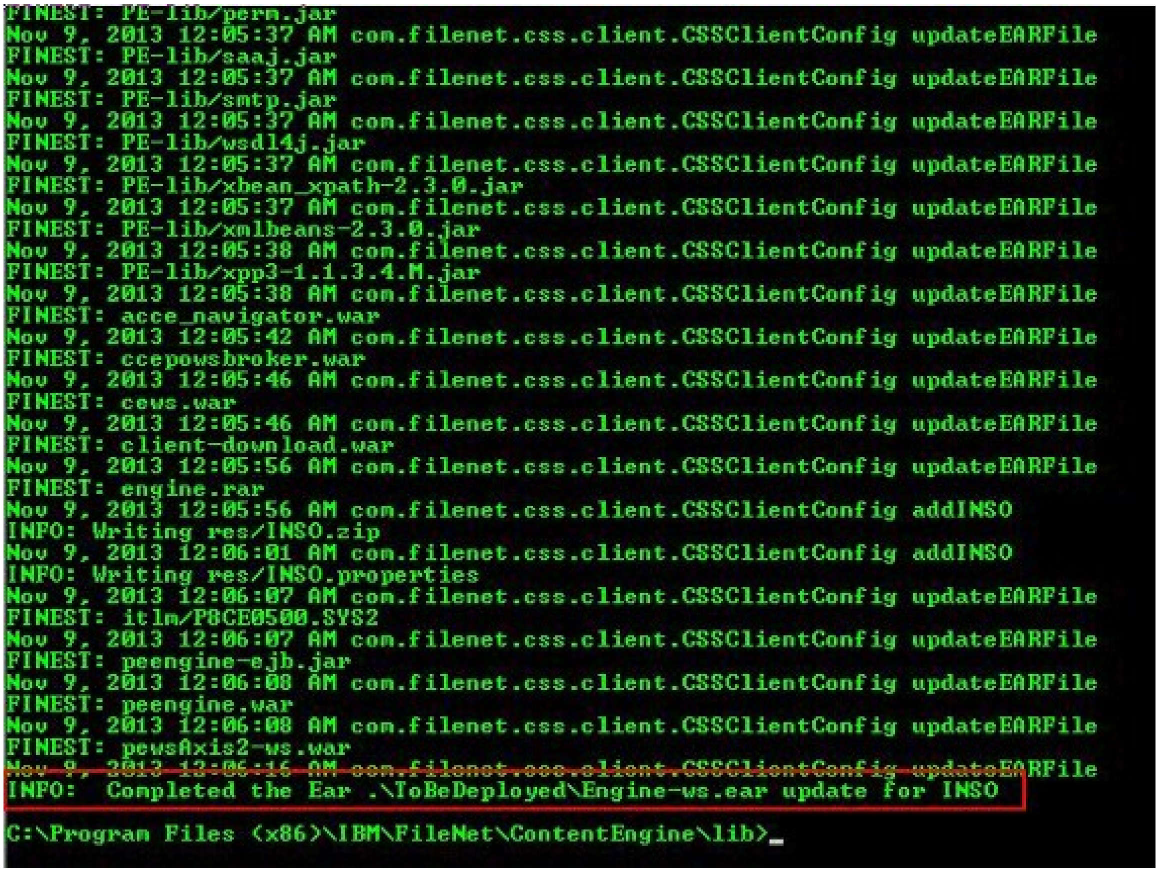
```
C:\Program Files (x86)\IBM\FileNet\ContentEngine\lib>java -cp "C:\Program Files (x86)\IBM\FileNet\Co
ntentEngine\lib\ToBeDeployed\Unzipped\commons-cli-1.0.jar";CSSClientConfig.jar com.filenet.css.clien
t.CSSClientTool -h

CEMP CSS Client Configuration Tool v1.0
Copyright (c) 2011-2012 IBM Corporation.  All Rights Reserved Worldwide.
usage: java -cp <commons-cli-1.X.jar;CSSClientConfig.jar> com.filenet.css.client.CSSClientTool [-i i
nso] [-d] [-v] [-h] [-c css] -e ear [-p platform] [-f]

  -p <platform>   Install CSS server client for platform. Supported <platform> options are:
                      (aix, hpux-ia, hpux-risc, zlinux, linux, solaris, win64)
  -c <css>        Install CSS server client. Where <css> is either true or false
  -d              Debug information logged
  -e <ear>        Target ear to be updated.  Where <ear> is the location (path\name) of the ear file

                      (Note that if the path contains spaces it must be enclosed in double quotes)
  -f              Force update of the Ear file
  -h              Prints usage information
  -i <inso>       Install INSO libraries. Where <inso> is either true or false
  -v              Version of the CSS Client and INSO versions
Example:
java -cp commons-cli-1.0.jar;CSSClientConfig.jar com.filenet.css.client.CSSClientTool -f -i true -e
"c:\Program Files\CSSClientConfigTool\Engine-ws.ear"
```

Figure 48: Running CSSClientTool



```
C:\Program Files (x86)\IBM\FileNet\ContentEngine\lib>java -cp "C:\Program Files (x86)\IBM\FileNet\Co
ntentEngine\lib\ToBeDeployed\Unzipped\commons-cli-1.0.jar";CSSClientConfig.jar com.filenet.css.clien
t.CSSClientTool -d -v -c false -e .\ToBeDeployed\Engine-ws.ear -p win64 -f
CSS API Version in EAR=2.1.0.0, 3740, 2013/01/17 17:42:35.704
INSO Version in EAR=null
INSO Platform in EAR=null
Installing CSS API Version in EAR =2.1.0.0, 3740, 2013/01/17 17:42:35.704
Installing INSO Version in EAR =8.3.7-p15954226_837.001


earPath=.\ToBeDeployed\Engine-ws.ear      debug=true      version=true      platform=win64      ecm
s=false      force = true


Nov 9, 2013 12:05:26 AM com.filenet.css.client.CSSClientConfig updateEARFile
INFO: Running the Update Ear Task for INSO  Platform=win64
Nov 9, 2013 12:05:26 AM com.filenet.css.client.CSSClientConfig updateEARFile
FINEST: ear INSO version  => null
Nov 9, 2013 12:05:26 AM com.filenet.css.client.CSSClientConfig updateEARFile
FINEST: ear INSO Platform  => null
Nov 9, 2013 12:05:26 AM com.filenet.css.client.CSSClientConfig updateEARFile
FINEST: ear cssAPIVersion  => 2.1.0.0, 3740, 2013/01/17 17:42:35.704
Nov 9, 2013 12:05:26 AM com.filenet.css.client.CSSClientConfig updateEARFile
FINEST: installingINSOVersion  => 8.3.7-p15954226_837.001
Nov 9, 2013 12:05:26 AM com.filenet.css.client.CSSClientConfig updateEARFile
```

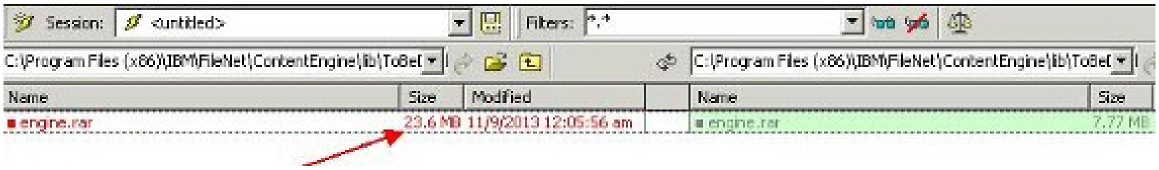
Figure 49: Completion status of CSSClientTool configuration



```
FINEST: PE-lib/perm.jar
Nov 9, 2013 12:05:37 AM com.filenet.css.client.CSSClientConfig updateEARFile
FINEST: PE-lib/saaj.jar
Nov 9, 2013 12:05:37 AM com.filenet.css.client.CSSClientConfig updateEARFile
FINEST: PE-lib/smtp.jar
Nov 9, 2013 12:05:37 AM com.filenet.css.client.CSSClientConfig updateEARFile
FINEST: PE-lib/wsdl4j.jar
Nov 9, 2013 12:05:37 AM com.filenet.css.client.CSSClientConfig updateEARFile
FINEST: PE-lib/xbean_xpath-2.3.0.jar
Nov 9, 2013 12:05:37 AM com.filenet.css.client.CSSClientConfig updateEARFile
FINEST: PE-lib/xmlbeans-2.3.0.jar
Nov 9, 2013 12:05:38 AM com.filenet.css.client.CSSClientConfig updateEARFile
FINEST: PE-lib/xpp3-1.1.3.4.M.jar
Nov 9, 2013 12:05:38 AM com.filenet.css.client.CSSClientConfig updateEARFile
FINEST: acce_navigator.war
Nov 9, 2013 12:05:42 AM com.filenet.css.client.CSSClientConfig updateEARFile
FINEST: ccepowsbroker.war
Nov 9, 2013 12:05:46 AM com.filenet.css.client.CSSClientConfig updateEARFile
FINEST: cews.war
Nov 9, 2013 12:05:46 AM com.filenet.css.client.CSSClientConfig updateEARFile
FINEST: client-download.war
Nov 9, 2013 12:05:56 AM com.filenet.css.client.CSSClientConfig updateEARFile
FINEST: engine.rar
Nov 9, 2013 12:05:56 AM com.filenet.css.client.CSSClientConfig addINSO
INFO: Writing res/INSO.zip
Nov 9, 2013 12:06:01 AM com.filenet.css.client.CSSClientConfig addINSO
INFO: Writing res/INSO.properties
Nov 9, 2013 12:06:07 AM com.filenet.css.client.CSSClientConfig updateEARFile
FINEST: itlm/P8CE0500.SYS2
Nov 9, 2013 12:06:07 AM com.filenet.css.client.CSSClientConfig updateEARFile
FINEST: peengine-ejb.jar
Nov 9, 2013 12:06:08 AM com.filenet.css.client.CSSClientConfig updateEARFile
FINEST: peengine.war
Nov 9, 2013 12:06:08 AM com.filenet.css.client.CSSClientConfig updateEARFile
FINEST: pewsAxis2-ws.war
Nov 9, 2013 12:06:16 AM com.filenet.css.client.CSSClientConfig updateEARFile
INFO:  Completed the Ear .\ToBeDeployed\Engine-ws.ear update for INSO

C:\Program Files (x86)\IBM\FileNet\ContentEngine\lib>_
```

Once done, the file INSO.zip is placed in the CPE EAR file. The following figures show its location inside the EAR file for versions prior to CPE 5.2.0.3, but the location of this zip file inside the EAR file would vary in future releases, for better handling.

Figure 50: engine.rar inside Engine-ws.jar (prior v5.2.0.3)



Session: <untitled>

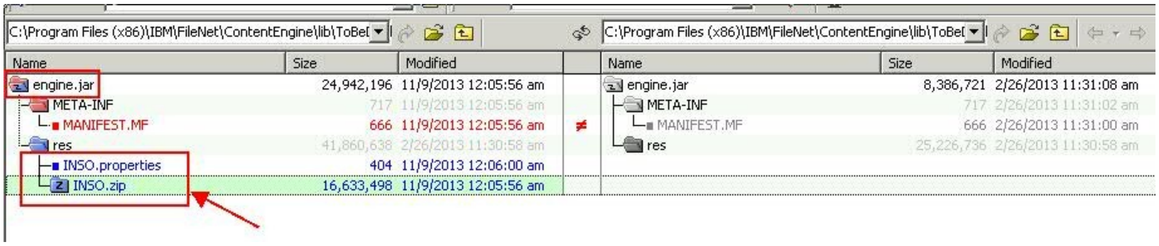
Filters: *.*

C:\Program Files (x86)\IBM\FileNet\ContentEngine\lib\ToBe(

C:\Program Files (x86)\IBM\FileNet\ContentEngine\lib\ToBe(

Name	Size	Modified		Name	Size
■ engine.rar	23.6 MB	11/9/2013 12:05:56 am		■ engine.rar	7.77 MB

Figure 51: Contents of engine.rar (prior v5.2.0.3)



C:\Program Files (x86)\IBM\FileNet\ContentEngine\lib\ToBe(

C:\Program Files (x86)\IBM\FileNet\ContentEngine\lib\ToBe(

Name	Size	Modified		Name	Size	Modified
engine.jar	24,942,196	11/9/2013 12:05:56 am		engine.jar	8,386,721	2/26/2013 11:31:08 am
META-INF	717	11/9/2013 12:05:56 am		META-INF	717	2/26/2013 11:31:02 am
■ MANIFEST.MF	666	11/9/2013 12:05:56 am	≠	■ MANIFEST.MF	666	2/26/2013 11:31:00 am
res	41,860,638	2/26/2013 11:30:58 am		res	25,226,736	2/26/2013 11:30:58 am
■ INSO.properties	404	11/9/2013 12:06:00 am				
INSO.zip	16,633,498	11/9/2013 12:05:56 am				

Starting CPE 5.2.0.3 and higher, for WebSphere/Weblogic, the INSO.zip file is located in the EAR under APP-INF\classes; and for JBoss it lies in the EAR under \lib but wrapped in a jar (INSO.jar) file.

Once the bootstrap and text search configuration steps are completed, the resultant “engine-xx.ear” file needs to be deployed in the Application Server.

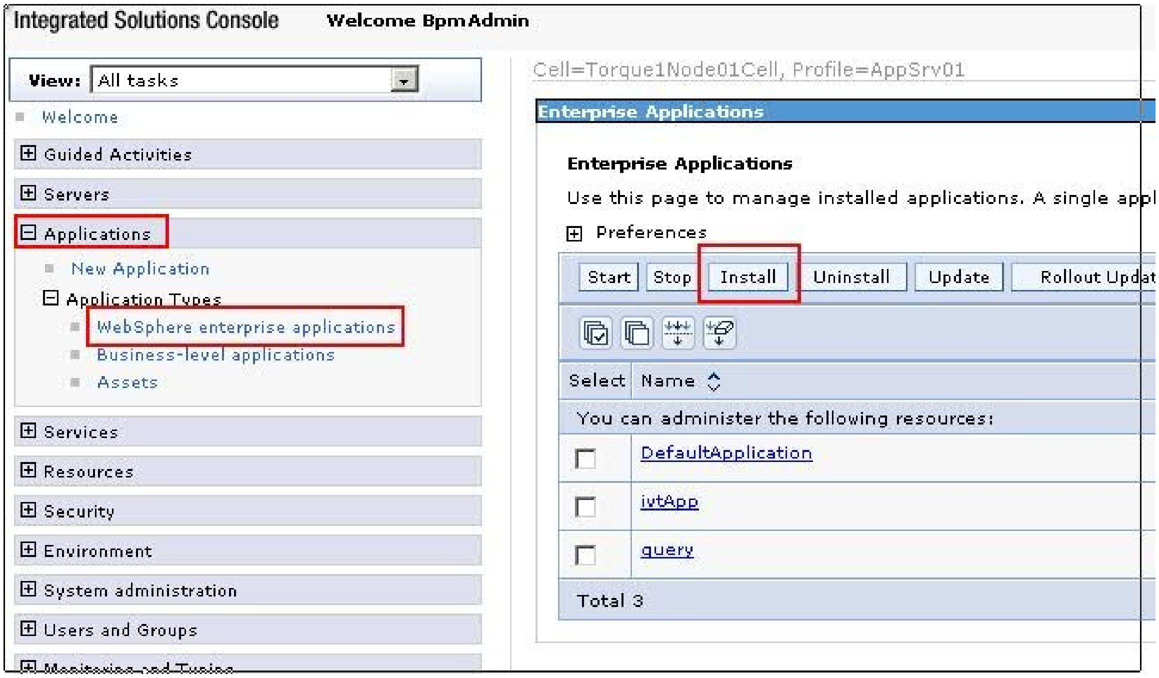
4. Deployment

The next and final step in Content Platform Engine configuration is to make the modified EAR (EAR that has been bootstrapped and been added with the CSS Client) available for use by a process called “Deployment” on the Application server.

Deploy CPE EAR as a “WebSphere Enterprise Application” by following the steps below:

Browse to the WebSphere enterprise applications node on the left and select the option "Install" to deploy the new application.

Figure 52: Install a New application



Cell=Torque1Node01Cell, Profile=AppSrv01

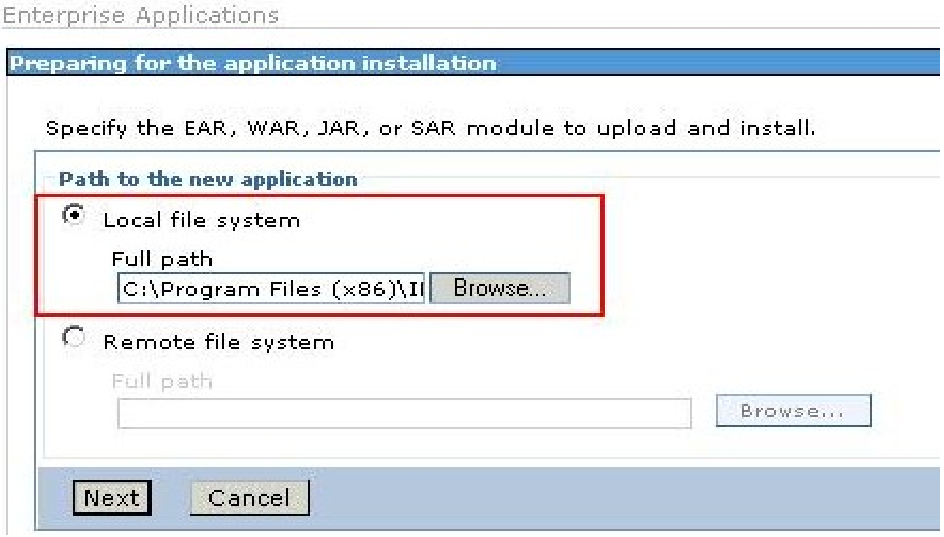
Enterprise Applications

Use this page to manage installed applications. A single appl

Select	Name
You can administer the following resources:	
□	DefaultApplication
□	ivtApp
□	query
Total 3	

Browse and choose the bootstrapped CPE EAR file for installation.

Figure 53: Select the local bootstrapped CPE EAR file "Engine-ws.ear"


Enterprise Applications
Preparing for the application installation
Path to the new application
Browse...
Browse...
Next
Cancel

Specify the EAR, WAR, JAR, or SAR module to upload and install.

● Local file system

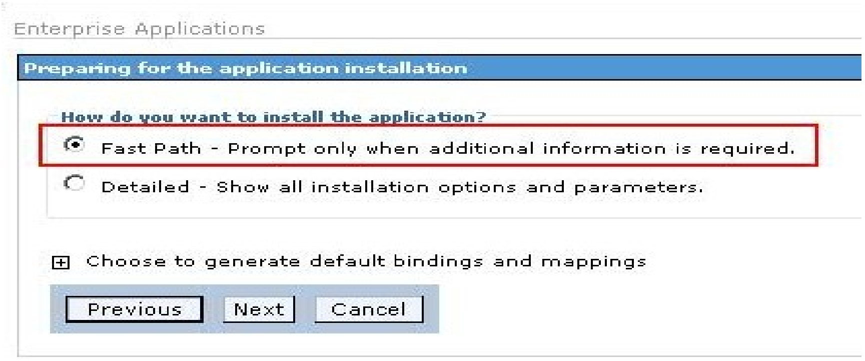
Full path C:\Program Files (x86)\II

○ Remote file system

Full path

Choose the fast path for deployment which would prompt only for any information that is required.

Figure 54: Select "Fast Path" for deployment



Enterprise Applications

Preparing for the application installation

How do you want to install the application?

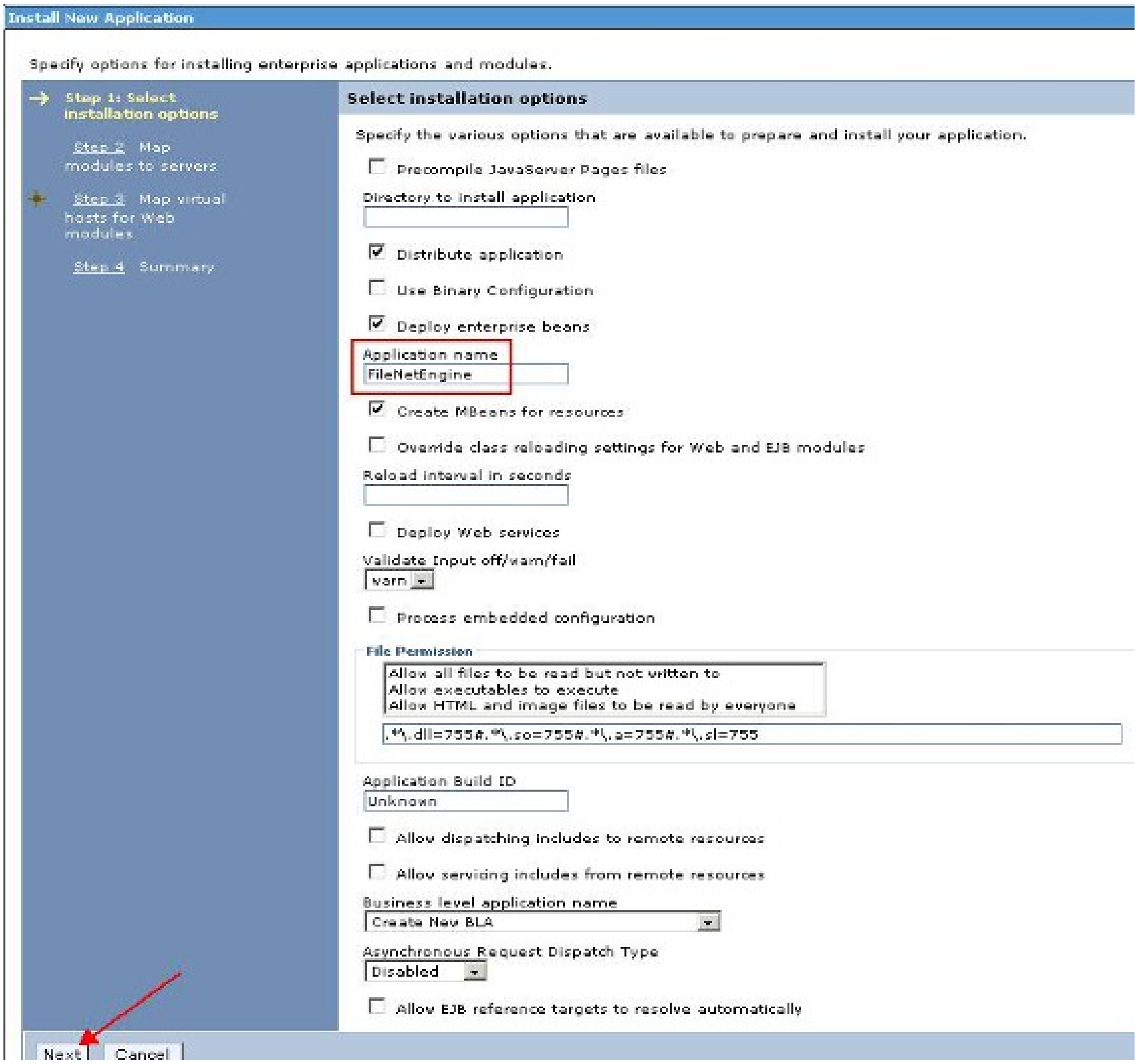
● Fast Path - Prompt only when additional information is required.

○ Detailed - Show all installation options and parameters.

⊞ Choose to generate default bindings and mappings

In the next page, the Application name would be defaulted to “FileNetEngine”. Go with the default options and click Next”

Figure 55: Verify installation options (enter “FileNetEngine” as Application Name if not already defaulted to)



Install New Application

Specify options for installing enterprise applications and modules.

→ Step 1: Select installation options

Step 2 Map modules to servers

Step 3 Map virtual hosts for Web modules

Step 4 Summary

Select installation options

Specify the various options that are available to prepare and install your application.

☐ Precompile JavaServer Pages files

Directory to install application:

☑ Distribute application

☐ Use Binary Configuration

☑ Deploy enterprise beans

Application name: FileNetEngine

☑ Create MBeans for resources

☐ Override class reloading settings for Web and EJB modules

Reload interval in seconds:

☐ Deploy Web services

Validate Input off/warn/fail: warn

☐ Process embedded configuration

File Permission

- Allow all files to be read but not written to
- Allow executables to execute
- Allow HTML and image files to be read by everyone

.*\.dll=755#.*\.so=755#.*\.a=755#.*\.sl=755

Application Build ID: Unknown

☐ Allow dispatching includes to remote resources

☐ Allow servicing includes from remote resources

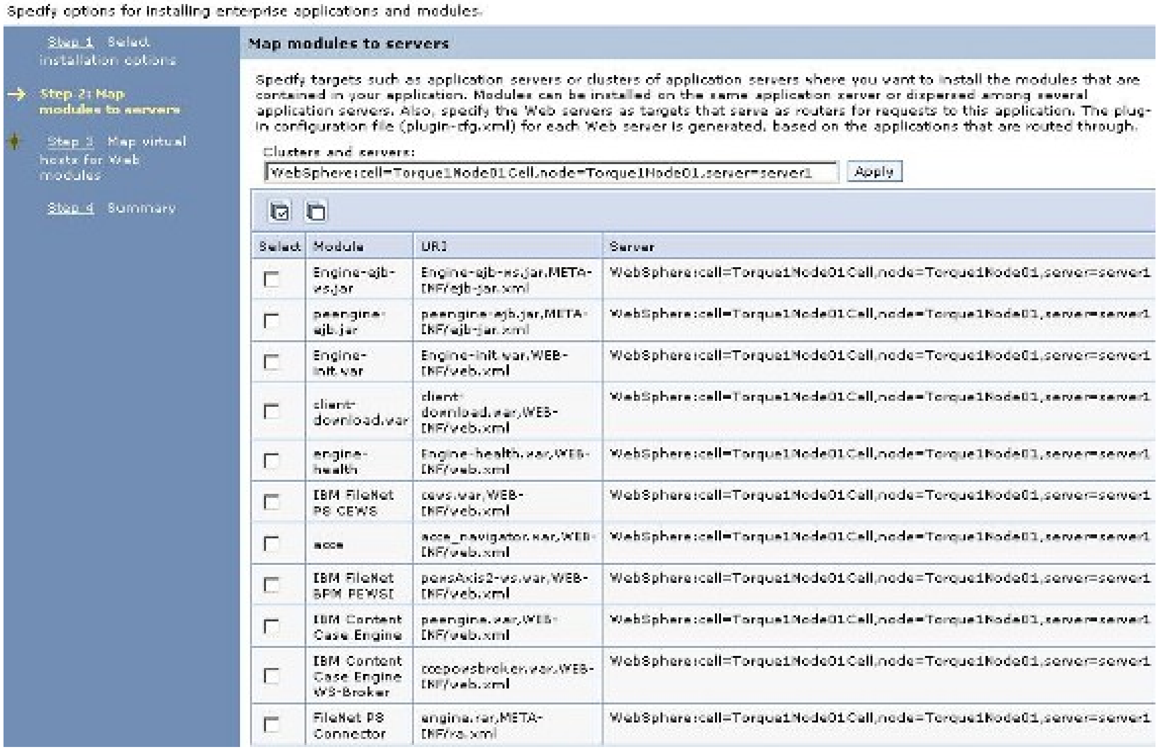
Business level application name: Create New BLA

Asynchronous Request Dispatch Type: Disabled

☐ Allow EJB reference targets to resolve automatically

Go with the default selections in the next wizard pages.

Figure 56: Map modules to servers



Specify options for installing enterprise applications and modules.

Step 1 Select installation options

→ Step 2: Map modules to servers

Step 3 Map virtual hosts for Web modules

Step 4 Summary

Map modules to servers

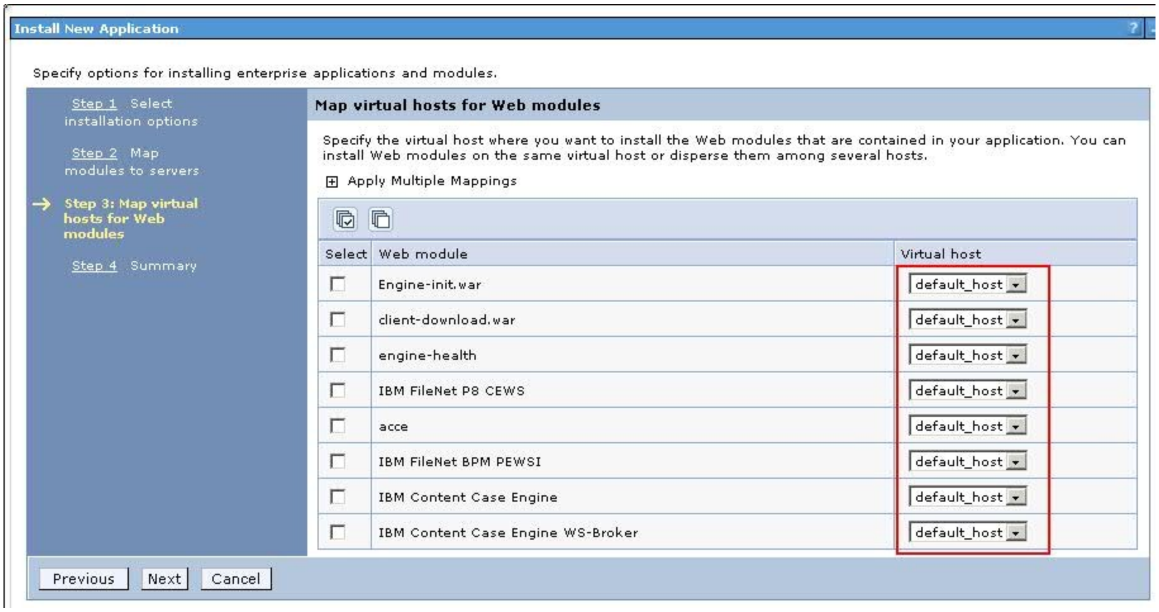
Specify targets such as application servers or clusters of application servers where you want to install the modules that are contained in your application. Modules can be installed on the same application server or dispersed among several application servers. Also, specify the Web servers as targets that serve as routers for requests to this application. The plug-in configuration file (plugin-cfg.xml) for each Web server is generated, based on the applications that are routed through.

Clusters and servers:

WebSphere:cell=Torque1Node01Cell,node=Torque1Node01,server=server1 Apply

Select	Module	URI	Server
□	Engine-ejb-ws.jar	Engine-ejb-ws.jar,META-INF/ejb-jar.xml	WebSphere:cell=Torque1Node01Cell,node=Torque1Node01,server=server1
□	peengine-ejb.jar	peengine-ejb.jar,META-INF/ejb-jar.xml	WebSphere:cell=Torque1Node01Cell,node=Torque1Node01,server=server1
□	Engine-init.war	Engine-init.war,WEB-INF/web.xml	WebSphere:cell=Torque1Node01Cell,node=Torque1Node01,server=server1
□	client-download.war	client-download.war,WEB-INF/web.xml	WebSphere:cell=Torque1Node01Cell,node=Torque1Node01,server=server1
□	engine-health	Engine-health.war,WEB-INF/web.xml	WebSphere:cell=Torque1Node01Cell,node=Torque1Node01,server=server1
□	IBM FileNet P8 CEWS	cews.war,WEB-INF/web.xml	WebSphere:cell=Torque1Node01Cell,node=Torque1Node01,server=server1
□	acce	acce_navigator.war,WEB-INF/web.xml	WebSphere:cell=Torque1Node01Cell,node=Torque1Node01,server=server1
□	IBM FileNet BPM PEWSI	pewsAxis2-ws.war,WEB-INF/web.xml	WebSphere:cell=Torque1Node01Cell,node=Torque1Node01,server=server1
□	IBM Content Case Engine	peengine.war,WEB-INF/web.xml	WebSphere:cell=Torque1Node01Cell,node=Torque1Node01,server=server1
□	IBM Content Case Engine WS-Broker	ccepowsbroker.war,WEB-INF/web.xml	WebSphere:cell=Torque1Node01Cell,node=Torque1Node01,server=server1
□	FileNet P8 Connector	engine.rar,META-INF/ra.xml	WebSphere:cell=Torque1Node01Cell,node=Torque1Node01,server=server1

Figure 57: Map virtual hosts for Web modules



Install New Application

Specify options for installing enterprise applications and modules.

Step 1 Select installation options

Step 2 Map modules to servers

→ Step 3: Map virtual hosts for Web modules

Step 4 Summary

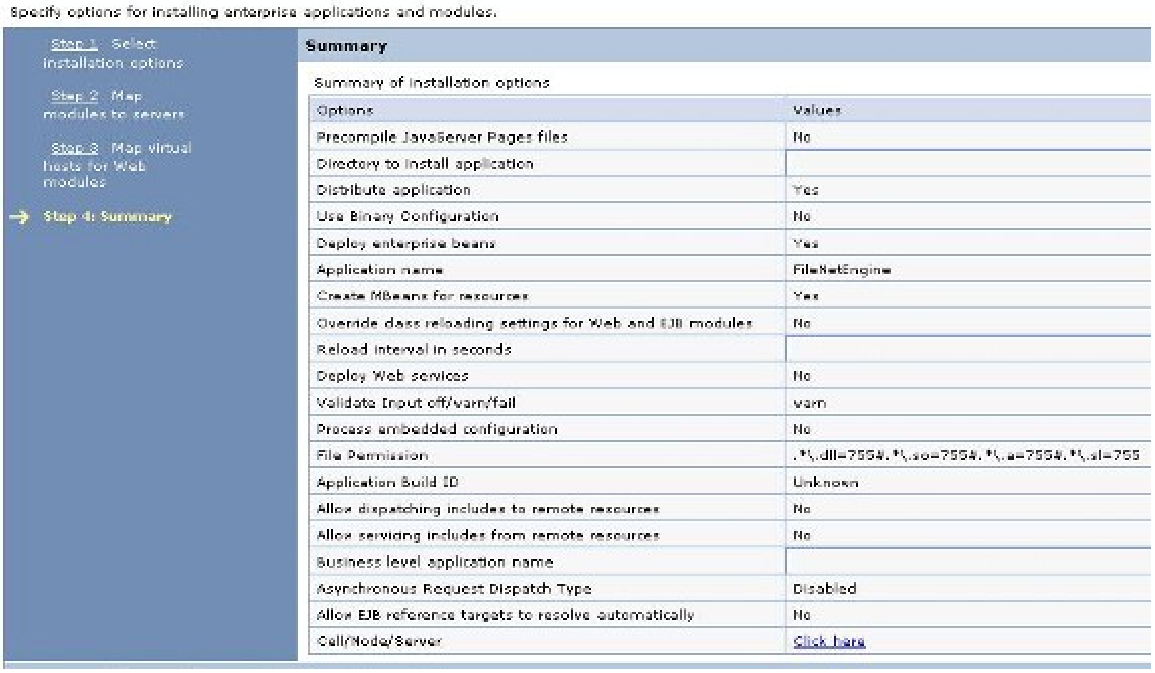
Map virtual hosts for Web modules

Specify the virtual host where you want to install the Web modules that are contained in your application. You can install Web modules on the same virtual host or disperse them among several hosts.

⊞ Apply Multiple Mappings

Select	Web module	Virtual host
□	Engine-init.war	default_host
□	client-download.war	default_host
□	engine-health	default_host
□	IBM FileNet P8 CEWS	default_host
□	acce	default_host
□	IBM FileNet BPM PEWSI	default_host
□	IBM Content Case Engine	default_host
□	IBM Content Case Engine WS-Broker	default_host

Figure 58: Summary of deployment



Step 1 Select installation options
Step 2 Map modules to servers
Step 3 Map virtual hosts for Web modules
Step 4: Summary

Specify options for installing enterprise applications and modules.

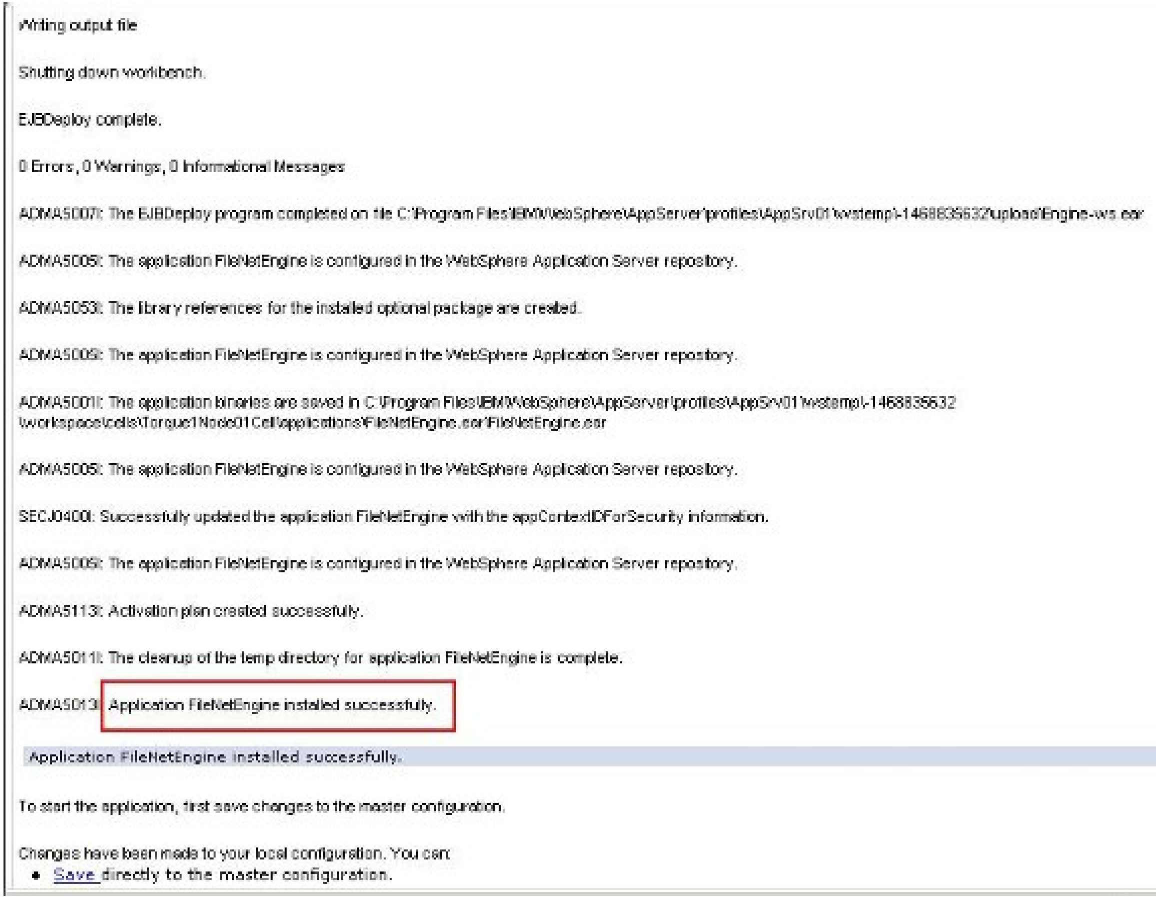
Summary

Summary of installation options

Options	Values
Precompile JavaServer Pages files	No
Directory to install application	
Distribute application	Yes
Use Binary Configuration	No
Deploy enterprise beans	Yes
Application name	FileNetEngine
Create MBeans for resources	Yes
Override class reloading settings for Web and EJB modules	No
Reload interval in seconds	
Deploy Web services	No
Validate Input off/warn/fail	warn
Process embedded configuration	No
File Permission	.*\.dll=755#.*\.so=755#.*\.a=755#.*\.sl=755
Application Build ID	Unknown
Allow dispatching includes to remote resources	No
Allow servicing includes from remote resources	No
Business level application name	
Asynchronous Request Dispatch Type	Disabled
Allow EJB reference targets to resolve automatically	No
Cell/Node/Server	Click here

The deployment can take several minutes. Once successful, select "Save directly to the master configuration" that would save the changes.

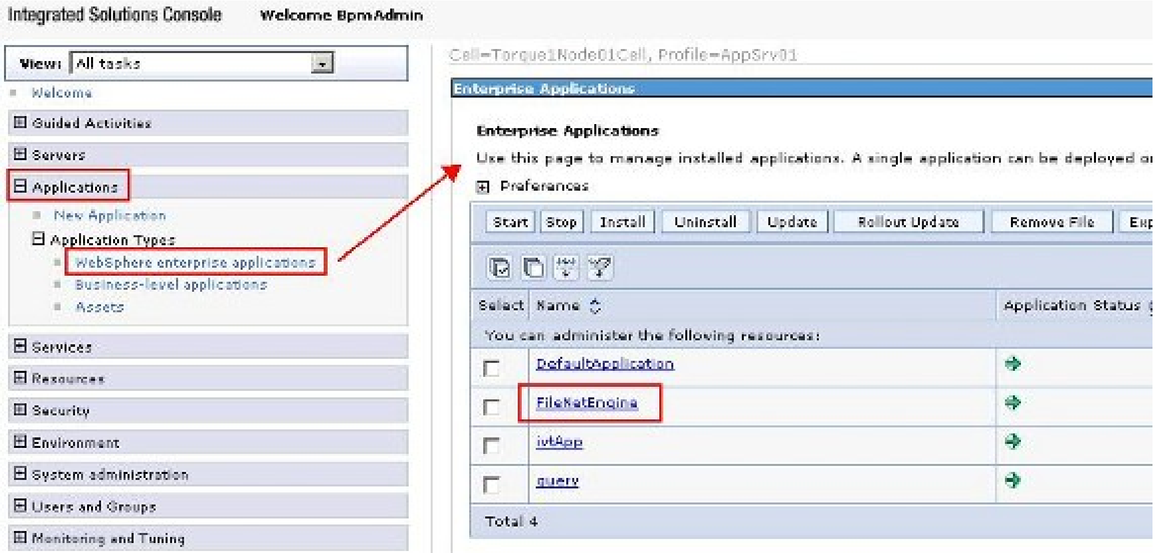
Figure 59: Successful Deployment



```
Writing output file

Shutting down workbench.

EJBDeploy complete.

0 Errors, 0 Warnings, 0 Informational Messages

ADMA5007I: The EJBDeploy program completed on file C:\Program Files\IBM\WebSphere\AppServer\profiles\AppSrv01\wstemp\-1468835632\upload\Engine-ws.ear

ADMA5005I: The application FileNetEngine is configured in the WebSphere Application Server repository.

ADMA5053I: The library references for the installed optional package are created.

ADMA5005I: The application FileNetEngine is configured in the WebSphere Application Server repository.

ADMA5001I: The application binaries are saved in C:\Program Files\IBM\WebSphere\AppServer\profiles\AppSrv01\wstemp\-1468835632
\workspace\cells\Torque1Node01Cell\applications\FileNetEngine.ear\FileNetEngine.ear

ADMA5005I: The application FileNetEngine is configured in the WebSphere Application Server repository.

SECJ0400I: Successfully updated the application FileNetEngine with the appContextIDForSecurity information.

ADMA5005I: The application FileNetEngine is configured in the WebSphere Application Server repository.

ADMA5113I: Activation plan created successfully.

ADMA5011I: The cleanup of the temp directory for application FileNetEngine is complete.

ADMA5013I: Application FileNetEngine installed successfully.

Application FileNetEngine installed successfully.

To start the application, first save changes to the master configuration.

Changes have been made to your local configuration. You can:
  • Save directly to the master configuration.
```

The application “FileNetEngine” is seen under the list of enterprise applications. (For upgrade scenarios, restart the WebSphere application server).

Figure 60: FileNetEngine application

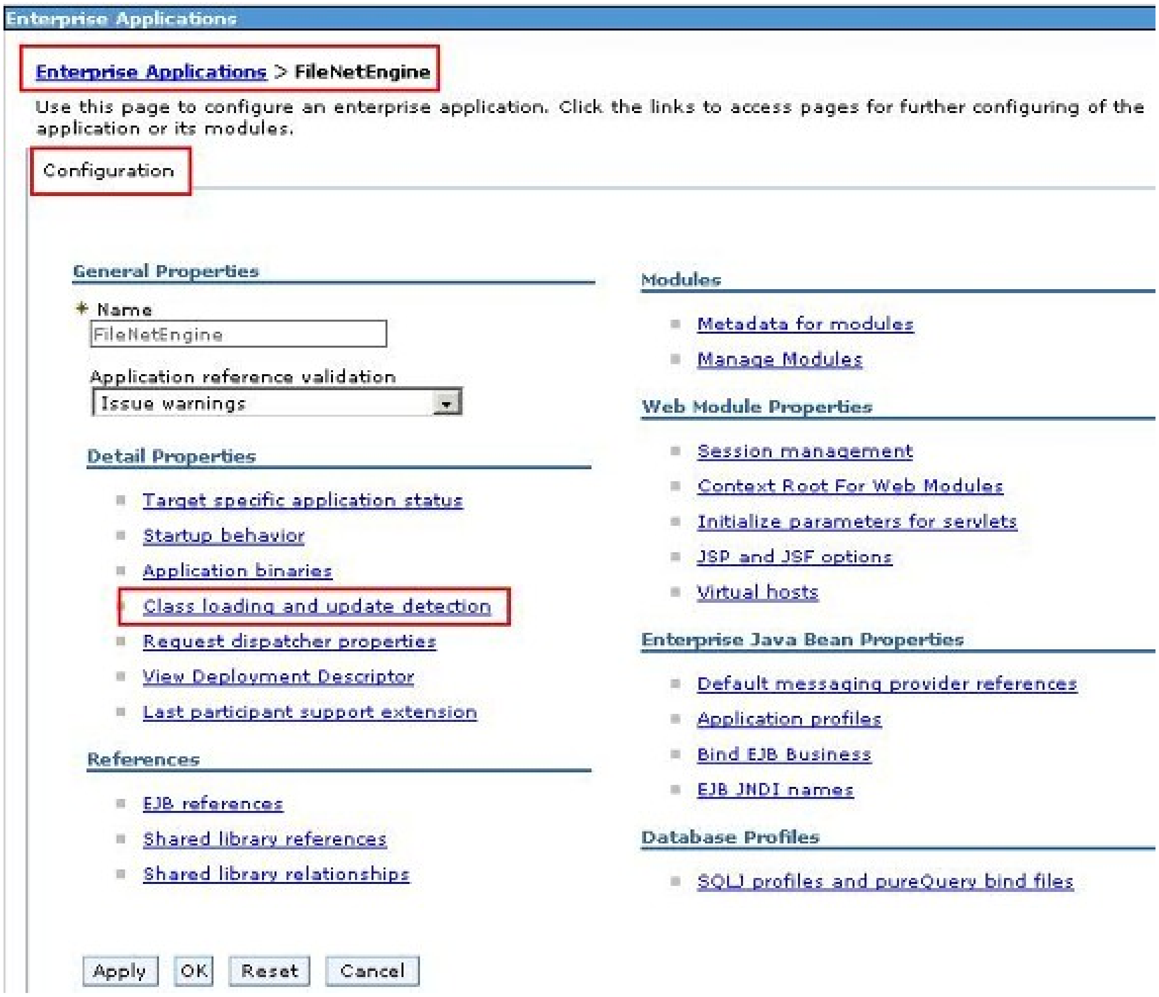


Use this page to manage installed applications. A single application can be deployed o

Select	Name	Application Status
You can administer the following resources:		
□	DefaultApplication	➜
□	FileNetEngine	➜
□	ivtApp	➜
□	query	➜
Total 4		

Content Platform Engine needs the classes for its application to be loaded first. So, the class loading options in application server should be set such that the class loader searches in the application class loader first rather than the parent class loader, to load a class.
Under the application, navigate to the class loading section and set (if not already set) the options appropriately as below:

Figure 61: Class loading and update detection configuration



Enterprise Applications

Enterprise Applications > FileNetEngine

Use this page to configure an enterprise application. Click the links to access pages for further configuring of the application or its modules.

Configuration

General Properties

* Name

FileNetEngine

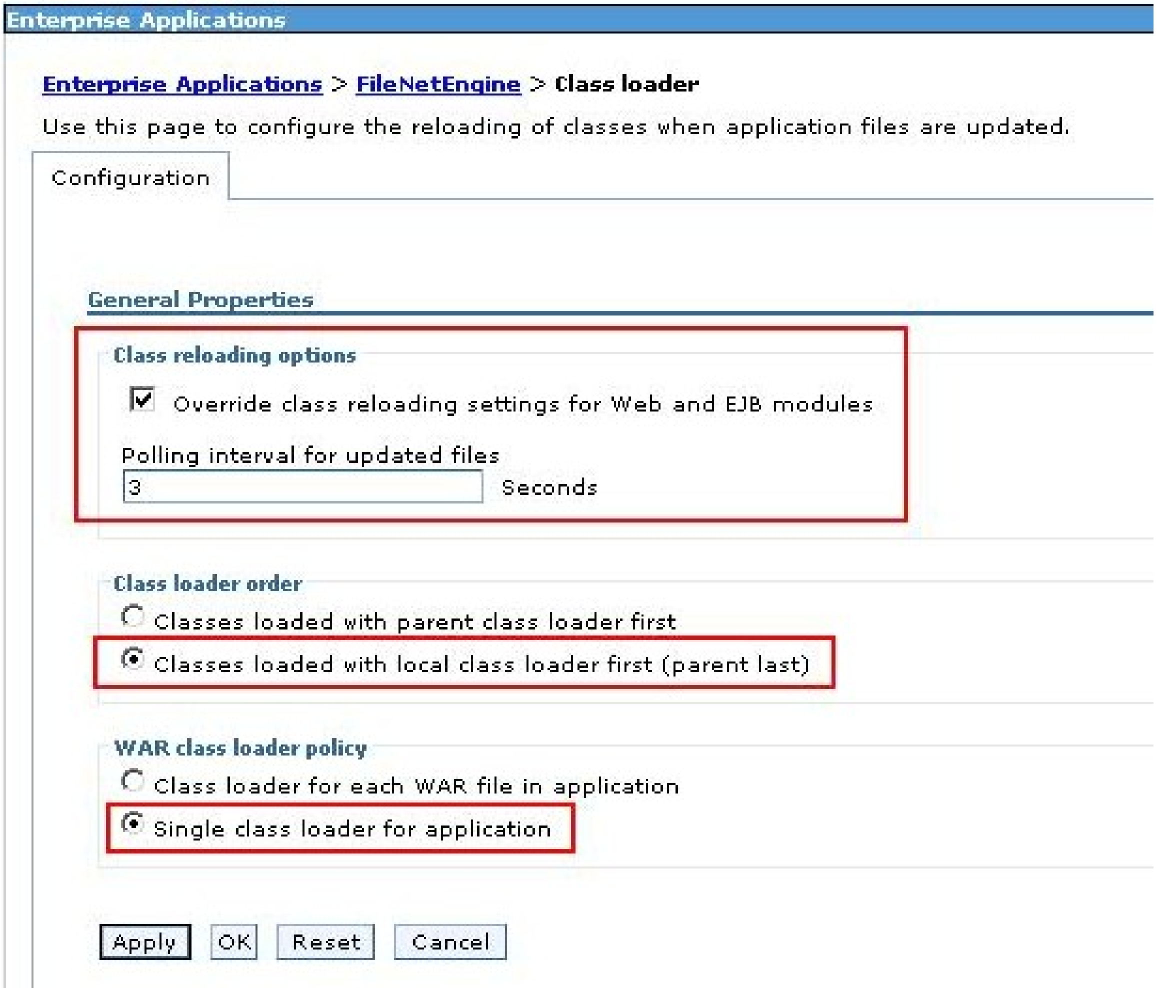
Application reference validation

Issue warnings

Detail Properties
Target specific application status
Startup behavior
Application binaries
Class loading and update detection
Request dispatcher properties
View Deployment Descriptor
Last participant support extension
References
EJB references
Shared library references
Shared library relationships
Modules
Metadata for modules
Manage Modules
Web Module Properties
Session management
Context Root For Web Modules
Initialize parameters for servlets
JSP and JSF options
Virtual hosts
Enterprise Java Bean Properties
Default messaging provider references
Application profiles
Bind EJB Business
EJB JNDI names
Database Profiles
SQLJ profiles and pureQuery bind files
Apply
OK
Reset
Cancel

Set “Override class reloading settings for Web and EJB modules”
Set the “Class loader order” to “Classes loaded with local class loader first (parent last)”;
Set “WAS class loader policy” to “Single class loader for application”

Figure 62: Setting the class loading properties



Enterprise Applications

Enterprise Applications > FileNetEngine > Class loader

Use this page to configure the reloading of classes when application files are updated.

Configuration

General Properties

Class reloading options

☑ Override class reloading settings for Web and EJB modules

Polling interval for updated files
3 Seconds

Class loader order

○ Classes loaded with parent class loader first
● Classes loaded with local class loader first (parent last)

WAR class loader policy

○ Class loader for each WAR file in application
● Single class loader for application

Once the options are set, restart the application server.

Open the browser and navigate the CPE ping page URL “http://server:port/FileNet/Engine” to view the details of Content Platform Engine running.

DISCLAIMER: This tutorial is purely for educational purposes. Although, it works for a CPE 5.2.x environment, the installation guides and other documentation should be referred in case the user faces any issue during configuration. The screen shots of WebSphere that are captured correspond to v7.0.0.27 for CPE 5.2.0 GA and they are shown for illustration purposes only. A newer version of WebSphere may see a slightly different UI.