

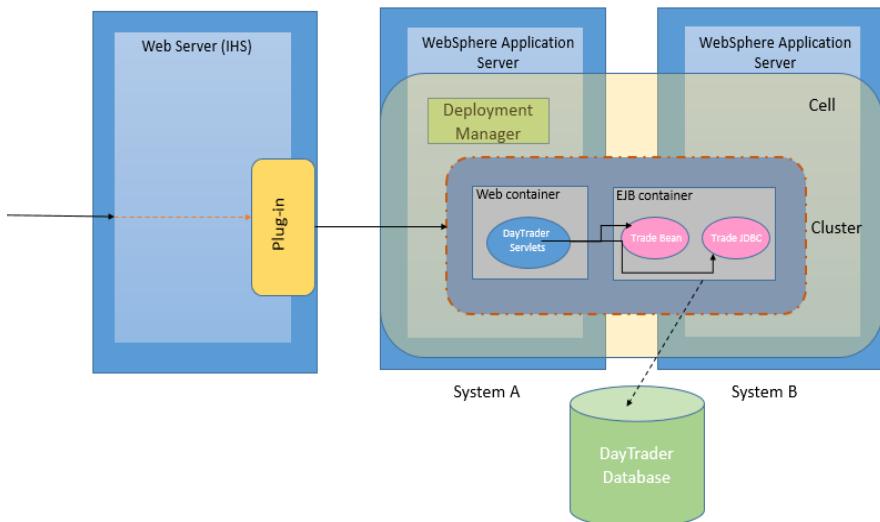
CHAPTER 10: APPLICATION DEPLOYMENT

Theory

WebSphere Application Server supports three different types of applications to be deployed that are:

- *Enterprise Applications*, that can be an EAR (Enterprise Archive), a WAR (Web Application) file or a JAR (Java Archive) file. When an EJB module or Web module is installed, WebSphere Application Server will automatically pack it as an EAR file. It is similar case for WAR files, meaning, deployment process will convert WAR file into EAR automatically.
- *Business Level Application*, is a WebSphere Application Server configuration artifact that provides a complete definition of an application from business view. It doesn't contain any application file but presents a configuration that contains units which represents application files.
- *Asset*, is an application binary file such as EAR files, EJB modules, web modules, shared libraries and etc. that are stored in a repository managed by WebSphere Application Server.

You can deploy the applications from local file system that means the file system of the system that deployment manager runs or from a remote file system which means your system that is connected to administrative console.



Before deploying an application, it is better to setup JDBC and JMS settings that will be used by the application. During installation of an application, you can set many options specific to your deployment.

One of the most important steps during the installation is mapping the application modules. You need to map the application to appropriate scope and also to your web server. This will ensure that with proper plug-in configuration, your server will be able to route the requests to your application.

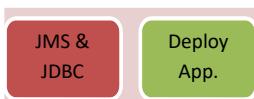
AIM

Day Trader is a benchmark application that simulates an online stock trading system. It was originally developed by IBM and later donated to the Apache Geronimo community. In this lab exercise, we will install Day Trader application and to achieve this goal, we need to follow 2 steps:

Task 1: Setup JMS and JDBC settings

Task 2: Deploy a sample application

Lab Exercise 10: APPLICATION DEPLOYMENT



-
1. **Setup JMS and JDBC settings**
 2. **Deploy a sample application**



Task 1: Setup JMS and JDBC Settings

Step 1: Navigate to “Security>Global security>JAAS-J2C authentication data” and click “New”.

The screenshot shows the WebSphere Integrate software interface. The left sidebar has a tree view with nodes like Welcome, Guided Activities, Servers, Applications, Jobs, Services, Resources, Runtime Operations, Security, and Global security. The Global security node is highlighted with a red box. The main panel shows a table titled "Global security > JAAS - J2C authentication data". The table has columns for Select, Alias, User ID, and Description. There is one entry: "None". At the bottom of the table, it says "Total 0". On the right side of the screen, there are several help links: "Field help", "More information about this page", "Command Assistance", and "Scripting command for last action".



Step 2: Add authentication alias for the database connection and administrative user.

Alias: MyWASDataSourceAuthData,

User ID: db2inst1 (Database user to run queries.)

Alias: MyWASAdminAuthData

User ID: ernesto (Administrative user of WebSphere Application Server)

Global security > JAAS - J2C authentication data > New...

Specifies a list of user identities and passwords for JavaTM 2 connector security to use.

General Properties

- + Alias: MyWASDataSource
- + User ID: db2inst1
- + Password: [REDACTED]
- Description: [REDACTED]

Apply | OK | Reset | Cancel |

Global security > JAAS - J2C authentication data > New...

Specifies a list of user identities and passwords for JavaTM 2 connector security to use.

General Properties

- + Alias: MyWASAdminAuthData
- + User ID: wasadmin
- + Password: [REDACTED]
- Description: [REDACTED]

Apply | OK | Reset | Cancel |



Step 3: Navigate to “Resources>JDBC>JDBC providers” and for the cell scope click “New”.

Step 4: Enter the database details. You can use the image below for DB2 configuration.



Step 5: Enter the path of the database drivers and then click "Next".

Create a new JDBC Provider

Step 1: Create new JDBC provider
Step 2: Enter database class path information
Step 3: Summary

Enter database class path information

Set the class path for the JDBC driver class files, which WebSphere(R) Application Server uses to define your JDBC provider. You can enter the directory path of one or more files or click the Add button to select multiple environment variables that define the directory locations of the files. Use complete directory paths when you type the [DB2] driver file locations. For example: C:\SQLLIB\java on Windows(R) or /home/ibm/db2inst1/sqllib\java on Linux(TM).

Entries are separated by using the ENTER key and must not contain path separator characters (such as ';' or ':'). If a value is specified for you, you may click Next to accept the value.

Class path:

```
$DB2UNIVERSAL_JDBC_DRIVER_PATH/db2jcc.jar  
$DB2UNIVERSAL_JDBC_DRIVER_PATH/db2jcc_license_cu.jar  
$DB2UNIVERSAL_JDBC_DRIVER_PATH/db2jcc_c_license_cisuz.jar
```

Directory location for "db2jcc.jar db2jcc_license_cisuz.jar" which is saved as WebSphere variable \$DB2UNIVERSAL_JDBC_DRIVER_PATH
/opt/IBM/DB2Drivers

Native library path
Directory location which is saved as WebSphere variable \$DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH

Previous Next > Cancel



Step 6: Click "Finish" to complete.

Screenshot of the WebSphere Integrations Console interface showing the 'Create a new JDBC Provider' wizard.

The left sidebar shows the navigation tree with sections like Applications, JMS, Services, Resources, JNDI, JDBC, Resource Adapters, Concurrency, Cache instances, Mail, URLs, and Resource Environment.

The main panel displays the 'Create a new JDBC Provider' wizard, Step 3: Summary. It shows the summary of actions:

Options	Values
Scope	cell:wasv90Cell01
JDBC provider name	DB2 Universal JDBC Driver Provider (XA)
Description	This class uses commit JDBC-JDBC 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 ZOS is not supported for data sources created under this provider.
Class path	\$DB2UNIVERSAL_JDBC_DRIVER_PATH\$ib2jcc.jar \$UNIVERSAL_JDBC_DRIVER_PATH\$ib2jcc_license_cu.jar \$DB2UNIVERSAL_JDBC_DRIVER_PATH\$ib2jcc_license_cisuz.jar \$DB2UNIVERSAL_JDBC_DRIVER_PATH\$ibpIBM0B2DDrivers \$UNIVERSAL_JDBC_DRIVER_PATH\$
Native path	\$DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH\$
Implementation class name	com.ibm.db2.jcc.DB2XADataSource

At the bottom of the wizard, there are three buttons: Previous, Finish (highlighted in red), and Cancel.



Step 7: Navigate to “Resources>JDBC>Data sources” and for the cell scope click “New”.

Select	Name	JNDI name	Scope	Provider	Description	Category
<input type="checkbox"/>	built-in derby-datasource	jbdc:built-in-derby-datasource	Cell=wasv90Cell01	Derby JDBC Provider (XA)	Derby is not supported in production	default

Step 8: Use “TradeDS” as a name and “jdbc/TradeDS” ad JNDI name and then click “Next”.



Step 9: Select the newly created JDBC provider and click “Next”.

Screenshot of the WebSphere Integrations console showing the "Create a data source" wizard, Step 2: Selected JDBC provider. The "Select an existing JDBC provider" radio button is selected, and "DB2 Universal JDBC Driver Provider (XA)" is chosen. The "Next" button is highlighted with a red box.

Step 10: Enter database specific properties. (database name is “mywasdb”).

Screenshot of the WebSphere Integrations console showing the "Create a data source" wizard, Step 3: Enter database specific properties for the data source. The "Driver type" is set to "4 -", "Database name" is "mywasdb", "Server name" is "was990", and "Port number" is "59000". The "Use this data source in container managed persistence (CMP)" checkbox is checked. The "Next" button is highlighted with a red box.



Step 11: Select “MyWASDataSourceAuthData” for authentication alias setting as below.

Cell-easy90Cell01, Profile-Dmgr01

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

Setup security aliases

Select the authentication values for this resource.

Authentication alias for XA resource
wasv90cellmanager01/MyWASDataSourceAuthData

Container-managed authentication alias
wasv90cellmanager01/MyWASDataSourceAuthData

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](#)

Previous Next Cancel



Step 12: Click "Finish" to complete.

Summary of actions:	
Options	Values
Scope	cells:FusionClouds_dmgrCell01
Data source name	TradeDataSource
JNDI name	jdbc/TradeDataSource
Select an existing JDBC provider	DB2 Universal JDBC Driver Provider Only (XA)
Implementation class name	com.ibm.db2.jcc.DB2XADataSource
Driver type	4
Database name	tradedb
Server name	db2.fusionclouds.com
Port number	50000
Use this data source in container managed persistence (CMP)	true
Authentication alias for XA recovery	TradeDataSourceAuthData
Component-managed authentication alias	TradeDataSourceAuthData
Mapping-configuration alias	(none)
Container-managed authentication alias	(none)

Previous **Finish** Cancel

Step 13: Repeat the same steps to add following data sources.

Data source Name:NoTxTradeDS

JNDI name: jdbc/ NoTxTradeDS

Data source Name:METradeDS

JNDI name: jdbc/ METradeDS



Screenshot of the WebSphere Integration console showing the 'Create a data source' wizard.

The 'Summary' step is displayed, showing the configuration details for the data source:

Options	Values
Scope	cells/wasv90Cell01
Data source name	NoTxTradeDS
JNDI name	jdbcNoTxTradeDS
Select an existing JDBC provider	DB2 Universal JDBC Driver Provider (XA)
Implementation class name	com.ibm.db2.jcc.DB2XADataSource

Below the table, there is a section titled 'Use this data source in container managed persistence (CMP)'. It contains the following entries:

Authentication alias for XA recovery	wasv90CellManager01.MyWASDataSourceAuthData
Component-managed authentication alias	wasv90CellManager01.MyWASDataSourceAuthData
Mapping configuration alias	(none)
Container-managed authentication alias	(none)

At the bottom of the wizard, there are three buttons: 'Previous', 'Finish', and 'Cancel'. The 'Finish' button is highlighted with a red box.



Step 14: Navigate to “Service integration>Buses” and click “New”.

The screenshot shows the 'Buses' list page in the WebSphere Integrate software. The left sidebar contains a navigation tree with various categories like 'Welcome', 'Guided Activities', 'Servers', etc. The main area displays a table of buses. A red box highlights the 'New...' button in the toolbar above the table. A large pink arrow points from the top right towards this red box.

Step 15: Enter name for the bus “WAS_Cluster” and mark “Bus security” then click “Next”.

The screenshot shows the 'Create a new Service Integration Bus' wizard. It has two steps: 'Step 1: Create a new bus' and 'Step 2: Confirm creation of new bus'. In Step 1, there is a text input field 'Enter the name for your new bus' containing 'WAS_Cluster', which is highlighted with a red box. There is also a checked checkbox 'Bus security' which is also highlighted with a red box. The 'Next' button is at the bottom of the step 1 panel.



Step 16: Click "Next" to continue.

This wizard assists you in securing your bus. In order to enable bus security you need to have administration security enabled for the cell the bus is defined in.

If this has not yet been done, clicking next will launch the Security Configuration Wizard for the cell. Once that has been configured that wizard will return to this one, so bus security can be configured.

Step 1: Create a new bus
Step 1.1: Introduction
Step 1.2: Specify transport level security
Step 1.3: Confirm the enablement of security
Step 2: Confirm creation of new bus

Previous | Next | Cancel

Step 17: Click "Next" to continue.

Enabling bus security ensures the following:

- Client applications can authenticate to the bus
- The authorization policy for the bus is enforced
- Peer messaging engines need to authenticate to each other

It does not ensure the confidentiality and integrity of the data, for this secure transports are required. This can be achieved either by having a secured network, or by requiring that clients, and messaging engines use encryption of their transports, i.e. SSL when communicating. In order to force clients to use encrypted transports the bus can be configured to require clients use SSL.

Require clients use SSL protected transports

Previous | Next | Cancel



Step 18: Select “Inherit the cell level security domain” and click “Next”.

The screenshot shows the "Configure bus security" wizard in the WebSphere Integrate... interface. The left sidebar lists various configuration categories like WebSphere, Applications, and Service Integration. The main panel is titled "Configure bus security" and displays "Step 1: Create a new bus". It asks to "Select the security domain for the bus" and provides three options: "Use the global security domain", "Inherit the cell level security domain" (which is selected and highlighted with a red box), and "Use an existing security domain". Below these options is a "Next" button, which is also highlighted with a red box. A status bar at the bottom indicates "Step 1 of 4: Configure security for the bus".



Step 19: Click "Next" to continue.

BusSecurityWizard.displayName

Configure security for the bus

Step 1: Create a new bus

Step 1.1: Introduction

Step 1.2: Specify transport level security

Step 1.3: Select the security domain for the bus

Step 1.4: Confirm the enablement of security

Step 2: Confirm creation of new bus

Confirm the enablement of security

The following is a summary of your selections. To complete the bus member creation, click Finish. If there are settings you wish to change, click Previous to review security settings.

Summary of actions to be performed based on the input provided.

Options	Values
Enable administrative security	Already configured prior to running this wizard
Enable bus security	True
Require use of SSL protected transports	True
Inter-engine Authentication alias	(none)
Bus security domain	Inheriting the cell level domain

Previous | Next | Cancel

Step 20: Click "Finish" to confirm create of the bus.

CreateBus.displayName

Create a new Service Integration Bus

Step 1: Create a new bus

Step 1.1: Introduction

Step 1.2: Specify transport level security

Step 1.3: Select the security domain for the bus

Step 1.4: Confirm the enablement of security

Step 2: Confirm creation of new bus

Confirm create of new bus

The following is a summary of your selections. To complete the bus creation, click Finish. If there are settings you wish to change, click Previous to review bus settings.

Summary of actions:

New bus "WAS Cluster" will be created with bus security setting "Enabled".

Previous | Finish | Cancel



Step 21: Navigate to “Service integration>Buses>WAS_Cluster” and click “Bus members” under “Topology”.

The screenshot shows the WebSphere Integrator software interface. On the left, there's a navigation tree with various categories like 'Guided Activities', 'Servers', 'Applications', etc. The main panel shows 'Buses > WAS_Cluster' configuration. A red box highlights the 'Topology' section, which includes fields for 'Name' (WAS_Cluster), 'UUID' (DECB0287AF27E86), and 'Description'. Below these are sections for 'Inter-engine transport chain', 'Discard messages', and 'Configuration reload enabled'. To the right, another red box highlights the 'Services' section, which lists 'Inbound services', 'Outbound rules', and 'WS-Notification services'. The top right corner has help links for 'Field help', 'Page help', and 'Command Assistance'.

Step 22: Click “Add”.

This screenshot shows the 'Buses > WAS_Cluster > Bus members' list page. A red box highlights the 'Add' button in the top-left of the list area. The table below shows one entry: 'None'. The top right corner has help links for 'Field help', 'Page help', and 'Command Assistance'.



Step 23: Select “Cluster” and “WAS_CLUSTER” then click “Next”.

The screenshot shows the 'WebSphere software' interface with a sidebar containing various navigation links. The main window displays a wizard titled 'Add bus member'. Step 1: Selected server, cluster or WebSphere MQ server. It asks to choose a server, cluster, or WebSphere MQ server to add to the bus. There are three options: 'Server' (selected), 'Cluster' (highlighted with a red box and 'WAS_CLUSTER' chosen from the dropdown), and 'WebSphere MQ server' (disabled). The 'Next' button at the bottom is also highlighted with a red box.

Step 24: Uncheck “Enable messaging engine policy assistance” and click “Next”.

The screenshot shows the 'WebSphere software' interface with a sidebar containing various navigation links. The main window displays a wizard titled 'Select Cluster Topology Pattern'. Step 1: Selected server, cluster or WebSphere MQ server. Step 2: Messaging engine policy assistance settings. It asks to select a predefined messaging engine policy to apply to the selected cluster when it is added as a bus member. A note says: 'Enabling messaging engine policy assistance enables a predefined or custom policy to be applied to the selected server cluster. Tooling will be enabled to assist in maintaining the policy if the server cluster changes in size. Restrictions will be placed on the changes that can be made to associated clusters.' A checkbox labeled 'Enable messaging engine policy assistance' is present, which is unchecked. The 'Next' button at the bottom is highlighted with a red box.



Step 25: Select “Data store” and click “Next”.

SSB0131.SelectMessageStoreType.displayName

Select message store type

Choose the type of message store for the persistence of message state

Step 1: Select server, cluster or WebSphere MQ service settings

Step 1.1: Select the type of message store

(The next step will depend on decisions made in the current step)

Step 1.1.1: Select the type of message store

File store Data store

Step 2: Summary

Previous Next Cancel

Step 26: Configure JNDI name as “jdbc/MEDDataSource” and select the authentication alias “TradeDataSourceAuthData” then click “Next”.

SSB0131.ConfigureDataSource.displayName

Configure the properties for a data store

Specify data store properties

Specify the properties for the data store

Step 1: Select server, cluster or WebSphere MQ service settings

Step 1.1: Selecting the type of message store

Step 1.1.1: Select the type of message store

Step 1.1.2: Specifying data store properties

Step 2: Summary

Data source (JNDI name): jdbc/MEDDataSource

Schema name: IBMSSIB

Authentication alias: wasv90CellManager01/MyWASDataSourceAuthData

Create tables

Restrict long running locks

Previous Next Cancel



Step 27: Select “Next” to continue.

The screenshot shows the 'Tune application server for messaging performance' step. The left sidebar lists various administrative tasks. The main panel shows configuration options for the Java Virtual Machine (JVM) settings. At the bottom, there are 'Previous', 'Next', and 'Cancel' buttons, with 'Next' being highlighted by a red box.

Step 28: Click “Finish” to complete.

The screenshot shows the 'Add a new bus member' step. The left sidebar lists various administrative tasks. The main panel shows configuration options for adding a new bus member. At the bottom, there are 'Previous', 'Finish', and 'Cancel' buttons, with 'Finish' being highlighted by a red box. A callout box points to the 'Summary' section, which lists actions like 'Adding server cluster "WAS_CLUSTER" as member of bus "WAS_Cluster"' and 'Data store settings:'. A red box highlights the 'Summary' section.



Step 29: Navigate to “System integration>Buses>WAS_Cluster>Bus members>Messaging engines” and click “Add messaging engine”.

Screenshot of the WebSphere Integrate console interface. The URL is https://wasy90:9043/bm/console/login.do?action=secure. The page title is "Buses > WAS_Cluster > Bus members > Messaging engines for WAS_CLUSTER". A red box highlights the "Add messaging engine" button. The left sidebar shows navigation categories like Welcome, Guided Activities, Servers, Applications, Jobs, Services, Resources, Runtime Operations, Security, Operational policies, Environment, System administration, Users and Groups, Monitoring and Tuning, Troubleshooting, Service Integration, Service mapping, Web services, and WS-notification. The bottom left corner shows a UDDI icon.

Step 30: Select “Data store” and click “Next”.

Screenshot of the WebSphere Integrate console interface. The URL is https://wasy90:9043/bm/console/login.do?action=secure. The page title is "Select message store type". A red box highlights the "Data store" option in the list. The left sidebar shows navigation categories like Welcome, Guided Activities, Servers, Applications, Jobs, Services, Resources, Runtime Operations, Security, Operational policies, Environment, System administration, Users and Groups, Monitoring and Tuning, Troubleshooting, Service Integration, Service mapping, Web services, and WS-notification. The bottom left corner shows a UDDI icon.

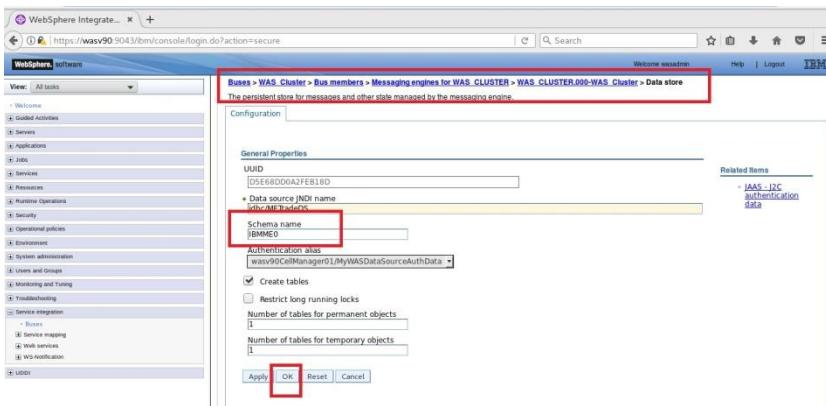


Step 31: Use “jdbc/MEDataSource” as JNDI name and select “TradeDataSourceAuthData” for authentication alias and click “Next”.

Step 32: Click “Finish” to complete.



Step 33: Navigate to “Service integration>Buses>WAS_Cluster>Bus members>Messaging engines for WAS_CLUSTER>WAS_CLUSTER.000-WAS_Cluster”, under additional properties Messaging Store opens up the Data store and change the schema as “IBMME0” then click “OK”.





Step 34: Navigate to “Servers>Core Groups>Core group settings” and click on “DefaultCoreGroup” and then click on “Policies” under “Additional properties”.

The image consists of two screenshots of the WebSphere Application Server administrative console. Both screenshots show the 'Core Groups' section of the 'Servers' guided activities.

Screenshot 1 (Top): This screenshot shows the 'Core Groups' page. A red box highlights the 'Core Groups' link in the left sidebar. Another red box highlights the 'DefaultCoreGroup' entry in the main table, which contains the following information:

Name:	DefaultCoreGroup
Description:	Default Core Group. The default core group cannot be deleted.
Total:	1

Screenshot 2 (Bottom): This screenshot shows the 'Core Groups > DefaultCoreGroup' configuration page. It has tabs for 'Runtime', 'Configuration', and 'Operations'. The 'Configuration' tab is active. A red box highlights the 'Policies' link under the 'Additional Properties' section. The 'Additional Properties' section also includes links for 'Core group servers', 'Discovery and coordination', 'Coordinator servers', and 'Custom properties'.



Step 35: Click on “New”.

Core Groups

Core Groups > DefaultCoreGroup > Policies

Use this page to view and manage the policies associated with a core group. Coordinators use these policies to determine on which servers the core group members are activated or deactivated.

Name	Description	Policy type	Match criteria
Clustered TM Policy	TM One-Of-N Policy	One of N policy	type=WSAF_TRANSACTIONS
Default SIB Policy	SIBus One-Of-N Policy	One of N policy	type=WSAF_SIB
Default SIP Quorum Policy	SIP All-active-policy with quorum disabled by default	All active policy	type=SIP_QUORUM

Total 3

Step 36: Select “One of N policy” and click “Next”.

Core Groups > DefaultCoreGroup > Policies > New...

Specifies the type of policy that you are creating.

Configuration

General Properties

Policies
[One of N policy](#)

Next | Cancel



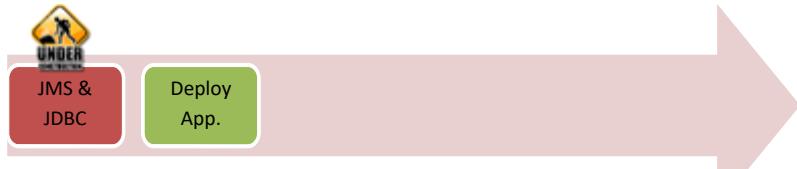
Step 37: Enter “Policy for MEO” as name, set “Is alive timer” to “30” and mark “Fallback” and “Preferred servers only” then click “OK”.

The screenshot shows the WebSphere Integration console interface. On the left, there's a navigation sidebar with various options like 'Welcome', 'Config Activities', 'Servers', 'Clusters', 'Database', 'Core Groups', 'Applications', 'Jobs', 'Services', 'Resources', 'Runtime operations', 'Security', 'Operational policies', 'Environment', 'System administration', and 'Users and Groups'. The 'Servers' section is currently selected. The main panel displays a configuration page for defining a 'One of N' policy. The 'General Properties' section includes fields for 'Name' (set to 'Policy for MEO'), 'Policy type' (set to 'One of N policy'), and a 'Description' field. Below this, under 'Additional Properties', there are checkboxes for 'Match criteria': 'Fallback' (checked) and 'Preferred servers only' (checked). There's also a 'Quorum' checkbox which is unchecked. At the bottom of the form are 'Apply', 'OK', 'Reset', and 'Cancel' buttons, with 'OK' being the one highlighted with a red box. A status message above the 'OK' button says: 'The additional properties will not be available until the configuration changes for this item are applied or saved.' To the right of the main panel, there's a help section with 'Page help' and 'More information about This page' links, along with a 'For field help' note about selecting field labels or list markers when the cursor is displayed.



Step 38: Navigate to “Core Groups>DefaultCoreGroup>Policies>Policy for MEO>Match criteria>” and add “WSAF_SIB_MESSAGING_ENGINE” as name and “WAS_CLUSTER.000-WAS_Cluster> as value, then click “OK”.

Step 39: Navigate to “Core Groups>DefaultCoreGroup>Policies>Policy for MEO>Preferred servers” and add the first member of the cluster, then “OK”.



Step 40: Repeat the same steps from 35 to 39 for the messaging engine “ME1” and for the second member of the cluster.

Step 41: Navigate to “System integration>Buses>WAS_Cluster>Security for bus WAS_Cluster>Users and groups in the bus connector role” and add administrative user “Ernesto” as name and “User” as type.

This screenshot shows the WebSphere Integration console interface. The left sidebar has sections like Servers, Applications, and Service integration. The main panel title is "Cell-wasv90Cell01, Profile-Dmgr01". It shows a table of users and groups under "Buses > WAS_Cluster > Security for bus WAS_Cluster > Users and groups in the bus connector role". The table has columns for Select, Name, Type, and Description. Two entries are listed: "Server" (Group) and "wasadmin" (User). A red box highlights the "New..." button in the toolbar above the table.

Step 42: Navigate to “Buses>WAS_Cluster>Destinations” and click “New”.

This screenshot shows the "Destinations" page for the "WAS_Cluster" bus. The left sidebar includes a "Service integration" section with "Destinations" selected. The main panel title is "Cell-wasv90Cell01, Profile-Dmgr01". It displays a table of destinations. A red box highlights the "New..." button in the toolbar above the table. The table columns are Identifier, Bus, Type, Description, and Mediation. Several entries are listed, including "Default Topic Space", "SYSTEM.Exception Destination WAS_CLUSTER.001-WAS_Cluster", and "SYSTEM.Exception Destination WAS_CLUSTER.001-WAS_Cluster". A red box also highlights the "Identifier" column header.



Step 43: Select “Queue” and then click “Next”.

The screenshot shows the 'Create new destination' dialog in the WebSphere Integrate console. On the left, a sidebar lists various management categories like Servers, Applications, and Security. The main panel is titled 'Create new destination' and contains the sub-instruction 'Create a new destination on this bus.'. Below this, a 'Select destination type' section has a radio button for 'Queue' which is selected (indicated by a red box). There are also options for 'Topic space', 'Alias', and 'Foreign'. At the bottom of this section is a 'Next' button, which is also highlighted with a red box.

Step 44: Use “TradeBrokerJSD” as identifier then click “Next”.

The screenshot shows the 'Set queue attributes' dialog in the WebSphere Integrate console. It's part of a three-step process: Step 1: Set queue attributes (which is active), Step 2: Assign the queue to a bus member, and Step 3: Confirm queue creation. In the 'Set queue attributes' step, there's a sub-instruction 'Configure the attributes of your new queue'. Under this, the 'Identifier' field is populated with 'TradeBrokerJSD', which is highlighted with a red box. Below the identifier field is a 'Description' input field containing 'Description'. At the bottom of the dialog is a 'Next' button, which is also highlighted with a red box. To the right of the dialog, a 'Help' section provides instructions for using the help feature.



Step 45: Select "Cluster=WAS_CLUSTER" as bus member and click "Next".

The screenshot shows the 'Create new queue' wizard in progress. The second step, 'Assign the queue to a bus member', has a red box around the 'Bus member' dropdown menu, which is set to 'Cluster=WAS_CLUSTER'. The 'Next' button at the bottom of the wizard is also highlighted with a red box.

Step 46: Click "Finish".

The screenshot shows the 'Create new queue' wizard in progress. The third step, 'Confirm queue creation', has a red box around the 'Finish' button at the bottom. A summary message in the center states: 'To complete creation of the queue, click Finish. If you want to change any selections, click Previous.' Below it, a 'Summary of actions:' section lists the new queue creation details.



Step 47: Navigate to “Buses>WAS_Cluster>Destinations>TradeBrokerJSD” and change “Quality of Service” settings as follows.

Default reliability: Express nonpersistent

Maximum reliability: Assured persistent.

The screenshot shows the WebSphere Integration console interface. On the left, there's a navigation tree with categories like Servers, Applications, Services, Resources, and System Administration. In the main panel, there's a form for configuring a destination. The 'Type' field is set to 'Queue'. Under 'Quality of Service', there are three dropdown menus: 'Default reliability' (set to 'Express nonpersistent'), 'Maximum reliability' (set to 'Assured persistent'), and 'Default priority' (set to '0'). A red box highlights the 'Quality of Service' section. To the right of the form, there are 'Related Items' links for 'Context properties', 'Mediation execution points', 'Application resources', and 'Topology'.



Step 48: Navigate to “Buses>WAS_Cluster>Destinations” and click “New”

The screenshot shows the 'Create new destination' dialog. On the left, there's a sidebar with various navigation options like Servers, Applications, Jobs, Services, Resources, etc. The main panel has a title 'Create new destination' and a sub-section 'Create a new destination on this bus.' It asks 'Select destination type' with three options: Queue (radio button), Topic space (radio button, highlighted with a red box), and Foreign (radio button). Below these options are 'Next' and 'Cancel' buttons, both of which are highlighted with red boxes.

Step 49: Use “Trade.Topic.Space” as identifier and click “Next”.

The screenshot shows the 'Set topic space Attributes' dialog. It has two main sections: 'Step 1: Set topic space Attributes' and 'Step 2: Confirm topic space creation'. In Step 1, there's a sub-section 'Configure the attributes of your new topic space' with a 'Identifier' field containing 'Trade.Topic.Space' (highlighted with a red box). There's also a 'Description' field with a large empty text area. At the bottom of Step 1 are 'Next' and 'Cancel' buttons, both highlighted with red boxes.



Step 50: Click “Finish” to complete.

Screenshot of the WebSphere administrative console showing the 'Confirm topic space creation' dialog box. The dialog box contains the following text:

- Create new topic space**
- Step 1: Set topic space Attributes**
- Step 2: Confirm topic space creation**
- Summary of actions:** New topic space "Trade.Topic.Space" will be created.
- Publication points for "Trade.Topic.Space" will be created on all bus members of bus "WAS_Cluster".**

At the bottom of the dialog box, there are three buttons: 'Previous', 'Finish' (which is highlighted with a red box), and 'Cancel'.

Step 51: Navigate to “Resources>JMS>Queue connection factories” and click “New”.

Screenshot of the WebSphere administrative console showing the 'Connection factories' list. The left sidebar shows the navigation path: Resources > JMS > Queue connection factories. The main panel displays a table of connection factories:

Select	Name	JNDI name	Provider	Description	Scope
<input type="checkbox"/>	built-in-jms-connectionfactory	jmsbuilt-in-jms-connectionfactory	Default messaging provider	Cell=wasv90Cell01	Cell=wasv90Cell01

At the top of the main panel, there is a toolbar with several icons. The 'New' button is highlighted with a red box. Other visible buttons include 'Delete', 'Edit', and 'Search'.



Step 52: Select “Default messaging provider” and click “OK”.

The screenshot shows the 'Connection factories' configuration page in the WebSphere Integration console. The left sidebar shows a tree structure with 'Servers', 'Applications', 'Services', and 'Resources' expanded. Under 'Resources', 'JMS' is selected, showing sub-options like 'JMSProviders', 'Connection factories', 'Queue connection factories', etc. The main panel shows a 'Select JMS resource provider' dialog. The 'Provider' dropdown is set to 'Default messaging provider'. The 'OK' button at the bottom of the dialog is highlighted with a red box.

Step 53: Use “TradeBrokerQCF” as name, “jms/TradeBrokerQCF” as JNDI name, select “WAS_Cluster” as bus member.

The screenshot shows the 'General Properties' configuration page for a JMS resource in the WebSphere Integration console. The left sidebar is identical to the previous screenshot. The main panel shows the 'General Properties' tab. Under 'Administration', the 'Name' field is set to 'TradeBrokerQCF' and the 'JNDI name' field is set to 'jms/TradeBrokerQCF'. In the 'Connection' section, the 'Bus name' dropdown menu is open, showing 'WAS_Cluster' as the selected option. Other options in the dropdown include 'JMS' and 'XA'. A note on the right side of the panel states: 'The additional properties will not take effect until the general properties for this item are applied or saved.'



Select “DefaultPrincipalMapping” as “Mapping-configuration alias” and click “OK”.

The screenshot shows the 'WebSphere Integration' interface with a sidebar containing 'Servers', 'Applications', 'Jobs', 'Services', and 'Resources' sections. Under 'Resources', the 'JMS' section is expanded, showing 'JMSSenders', 'Object pool managers', 'Java EE default resources', 'JMSProviders', 'Connection factories', 'Queue connection factories', 'Topic connection factories', 'Topics', and 'Activation specifications'. The 'JMSProviders' section is selected. On the right, there are several configuration tabs: 'Basic', 'Advanced', 'Administrative', 'Security settings', and 'Custom'. The 'Administrative' tab is active, showing options like 'Log missing transaction contexts', 'Manage cached handles', and 'Share data source with CMP'. The 'Security settings' tab is also visible, showing 'Authentication alias for XA recovery' set to 'None' and a dropdown for 'Mapping-configuration alias' which has 'DefaultPrincipalMapping' selected. At the bottom, there are 'Apply', 'OK', and 'Cancel' buttons, with 'OK' being the one highlighted with a red box.



Step 54: Navigate to “Resources>JMS>Topic connection factories” and click “New”.

The screenshot shows the WebSphere Integration console interface. The left sidebar has sections for Servers, Applications, Jobs, Services, Resources, JMS, JNDI, Resource Adapters, Concurrency, Cache Instances, Mail, URLs, and Resource Environment. The 'Resources' section is expanded, showing Subscribers, Object pool managers, Java EE default resources, JMS providers (Connection factories, Queue connection factories, Topic connection factories, Queues, Topics, Activation specifications), and JNDI. The 'JMS' section is also expanded. The main content area is titled 'Topic connection factories' with the sub-section 'Topic connection factories'. It explains what a topic connection factory is used for. A dropdown menu 'Scope' is set to 'All scopes'. Below it is a table with columns: Select, Name, JNDI name, Provider, Description, and Scope. The table shows one entry: 'None' under 'Name' and 'Total 0' under 'Count'. On the right side, there is a 'Help' panel with 'Field help' and 'Page help' sections.

Step 55: Select “Default messaging provider” and click “OK”.

This screenshot shows a modal dialog box titled 'Topic connection factories > Select JMS resource provider'. It has a 'Scope' dropdown set to 'cells:wasv90Cell01'. The main content area says 'Select the provider with which to create the Topic connection factory. The following providers support the selected resource type and are available at the selected scope.' There are two radio buttons: 'Default messaging provider' (selected) and 'WebSphere MQ messaging provider'. At the bottom are 'OK' and 'Cancel' buttons, with 'OK' highlighted by a red box.



Step 56: Use “TradeStreamerTCF” as name, “jms/TradeStreamerTCF” as JNDI name and select “WAS_Cluster” as bus name and click “OK”.

The screenshot shows the 'Administration' section of the JMS provider configuration. The 'Name' field is highlighted with a red box. The 'JNDI name' field is also highlighted with a red box. The 'Bus name' dropdown menu is open, showing 'WAS_Cluster' selected. Other options in the dropdown include 'Target' and 'Bus member name'.

Step 57: Navigate to “Resources>JMS>Queues” and click “New”.

The screenshot shows the 'Queues' table in the 'Resources > JMS > Queues' section. A red box highlights the 'New' button in the top-left corner of the table. The table has columns for 'Selected', 'Name', 'JNDI name', 'Provider', 'Description', and 'Scope'. The 'Name' column currently contains 'None'.



Step 58: Select “Default messaging provider” and click “OK”.

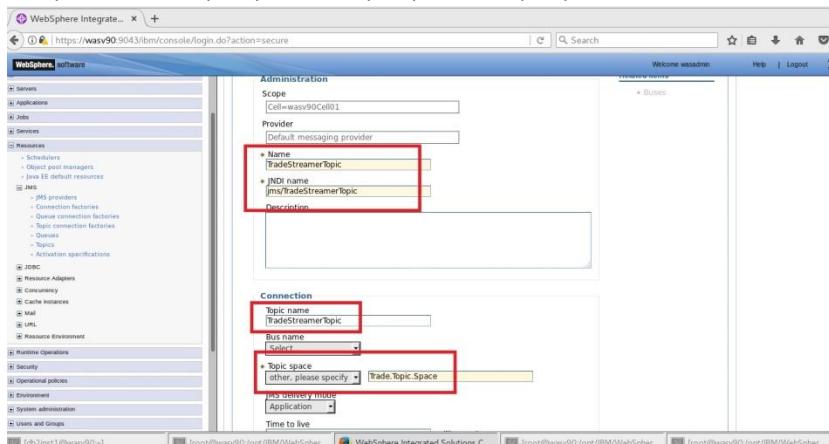
WebSphere Integrat... X +
 https://wasv90:9043/bm/console/login.do?action=secure
 WebSphere software
 Queues > Select JMS resource provider
 Scope [Cell:wasv90Cell01]
 Select the provider with which to create the Queue. The following providers support the selected resource type and are available at the selected scope.
 Default messaging provider
 WebSphere MQ messaging provider
 OK Cancel

Step 59: Use “TradeBrokerQueue” as name “jms/TradeBrokerQueue” as JNDI name, select “other, please specify” and enter “TradeBrokerJSD” as queue name and click “OK”.

WebSphere Integrat... X +
 https://wasv90:9043/bm/console/login.do?action=secure
 WebSphere software
 Scope [Cell:wasv90Cell01]
 Provider
 Default messaging provider
 Name TradeBrokerQueue
 JNDI name jms/TradeBrokerQueue
 Description
 Connection
 Bus name Local...
 Queue name other, please specify TradeBrokerJSD
 Delivery mode Application
 Priority



Step 60: Navigate to “Resources>JMS>Topics”, click “New”. Use “TradeStreamerTopic” as name, “jms/TradeStreamerTopic”, “TradeStreamerTopic” as topic name, and specify “Trade.Topic.Space” as topic space and click “OK”.





Step 61: Navigate to “Resources>JMS>Activation specification” and click “New”.

Step 62: Select “Default messaging provider” and click “OK”.



Step 63: Use “TradeBrokerMDB”, “eis/TradeBrokerMDB” as JNDI name, select “Queue” as destination and “jms/TradeBrokerQueue” as destination JNDI name and “WAS_Cluster” as bus name.

Step 64: Use “Subscription Durability” as follows.

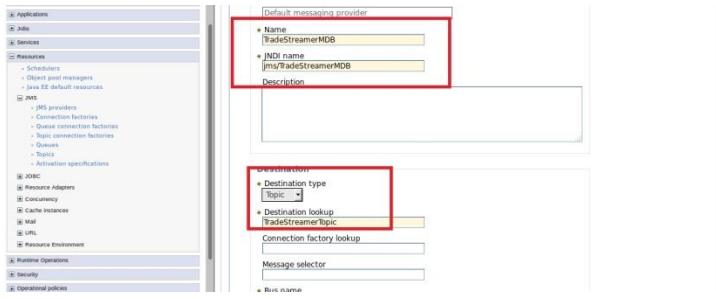


Step 65: Use “MyWASAdminAuthData” as authentication alias and click “OK”.



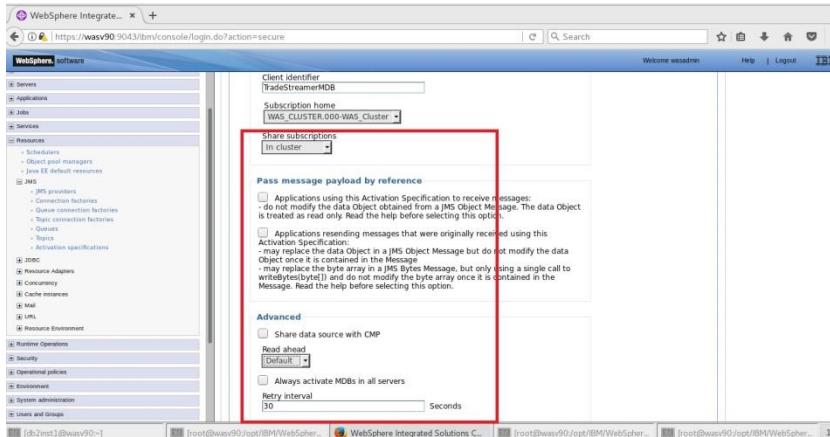
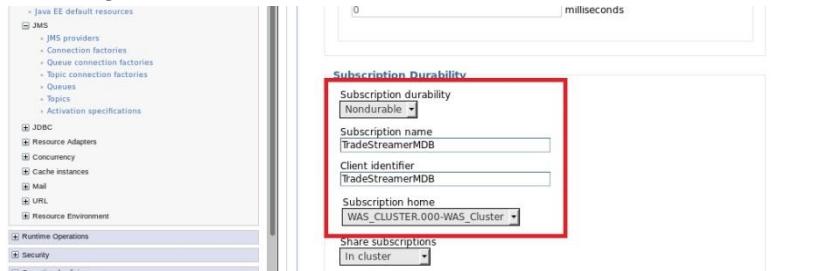


Step 66: Navigate to “Resources>JMS>Activation specification”, click “New”, Select “Default messaging provider” then click “OK”. Use “TradeStreamerMDB” as name, “eis/TradeStreamerMDB” as JNDI name, select “Topic” as destination and “jms/TradeStreamerTopic” as destination JNDI name, select “WAS_Cluster” as bus name.





Step 67: Configure “Subscription Durability”, “Advanced” and “Security settings” like in the images below and click “OK”.



Task 1 is complete!



Task 2: Deploy a sample application

Step 1: Navigate to “Applications>New Application” and click on “New Enterprise Application”.

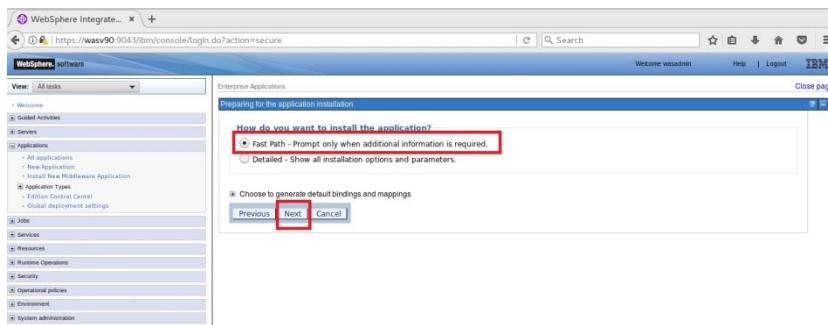
The screenshot shows the WebSphere software interface. On the left, there's a navigation sidebar with various options like 'Welcome', 'Guided Activities', 'Services', 'Applications', 'Jobs', 'Services', 'Resources', etc. The 'Applications' section is expanded, showing 'All applications', 'New Application', 'Install New Midware Application', 'Edition Control Center', and 'Global deployment settings'. On the right, a 'New Application' dialog box is open. It has a title 'New Application' and a sub-section 'New Application'. Below that, it says 'This page provides links to create new applications of different types.' There are three options: 'New Enterprise Application' (which is selected and highlighted with a red box), 'New Business Level Application', and 'New Asset'. At the bottom of the dialog, there are 'Cancel' and 'Next' buttons.

Step 2: Locate the ear file on the local file system and click “Next”.

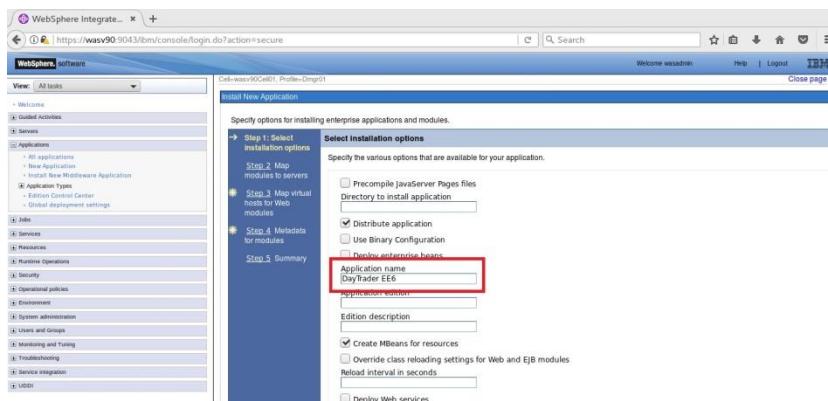
This screenshot shows the 'Preparing for the application installation' step. The left sidebar is identical to the previous screenshot. The main area has a title 'Preparing for the application installation' and a sub-section 'Specify the EAR, WAR, JAR, or SAR module to upload and install.' It contains a 'Path to the new application' field. There are two radio button options: 'Local file system' (unchecked) and 'Remote file system' (checked). Under 'Remote file system', there's a 'Full path' input field containing the path 'froot/software/LinuxApp/DayTrader3/install/DayTrader3-EE6/daytrader'. Below the path field are 'Browse...' and 'Browse... (local)' buttons. At the bottom of the dialog are 'Next' and 'Cancel' buttons. A 'Help' section on the right provides information about the 'Path to the new application' field.



Step 3: Select “Fast Path” and click “Next”.



Step 4: Change application name to “DayTrader2-EE6” and click “Next”.





Step 5: Select cluster and the web server to map both of the modules of the application and click “Apply”, then click “Next”.

Selected	Module	URI	Server
<input type="checkbox"/>	DayTrader	dt-ejb.jar.META-INF/ejb-jar.xml	WebSphere cell=wasv90Cell01,node=wasv90Node01,server=server1
<input checked="" type="checkbox"/>	Enterprise		
<input checked="" type="checkbox"/>	RestDefinitions		
<input checked="" type="checkbox"/>	DayTrader	web.war.WEB-INF/web.xml	WebSphere cell=wasv90Cell01,node=wasv90Node01,server=server1
<input type="checkbox"/>	Rest	Rest.war.IWEB-INF/web.xml	WebSphere cell=wasv90Cell01,node=wasv90Node01,server=server1
<input type="checkbox"/>	Rest	Rest.war.IWEB-INF/web.xml	WebSphere cell=wasv90Cell01,node=wasv90Node01,server=server1



Step 6: Click “Next”.

Specify options for installing enterprise applications and modules.

Step 4: Metadata for modules

The metadata-complete attribute defines whether the deployment descriptor for this module is complete. Set the metadata-complete attribute to "true" to merge and persist annotation-based metadata with existing XML-based deployment descriptor metadata to avoid scanning of annotation-based metadata each time the module is read. If the attribute remains "false", then the annotation-based metadata is scanned each time the module is read and can impact performance.

Module	URI	metadata-complete attribute
DayTrader Enterprise Bean Definitions	dt-ejb.jar META-INF/ejb-jar.xml	<input type="checkbox"/>
DayTrader Web	web.war.WEB-INF/web.xml	<input type="checkbox"/>
Rest.war	Rest.war.WEB-INF/web.xml	<input type="checkbox"/>

Step 5: Summary

Previous Next Cancel

Step 7: Click “Finish” to start deployment.

Step 5: Summary

Application name	DayTrader EE6
Application edition	
Edition description	
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 when fail	warn
Process embedded configuration	No
File Permission	\".dll=7558 \".so=7558 \".a=7558 \".\al=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
Deploy client modules	No
Client deployment mode	Isolated
Validate schema	No
Cell/Node/Server	Click here

Previous Finish Cancel



Step 8: You should see the success message, then click “Save” to write changes to the master configuration.

WebSphere Integrat... IBM

Welcome Logout

View: All tasks

Applications

- All applications
- New applications
- Imported from WebSphere Application
- Application Types**
- Edit Configuration Center
- Global deployment settings

Jobs

Services

Problems

Runtime Operations

Security

Operational policies

Environment

System administration

Users and Groups

Monitoring and Testing

Troubleshooting

Service Integrator

LDDI

https://wasv90:9043/bm/console/login.do?action=secure

ADM40113W: Resource assignment of file docroot/radical/DayTrader/DayTrader.war type java耳jarWARresource with And name pcom/radical/daytrader is not found within scope of module DayTrader.war with identifier was.WEB-INF\wabs\vm deployed to target WebSphereCell\wasCell02\cluster\WAS_CLUSTER.

ADM40081: The resource validation for application DayTrader EEE completed successfully, but warnings occurred during validation.

ADM40051: Application and module versions are validated with versions of deployment targets.

ADM40051: The application DayTrader EEE is configured in the WebSphere Application Server repository.

ADM40051: The application DayTrader EEE is configured in the WebSphere Application Server repository.

ADM40051: The bootstrap address for client module is configured in the WebSphere Application Server repository.

ADM40051: The library references for the installed optional package are created.

ADM40051: The application DayTrader EEE is configured in the WebSphere Application Server repository.

ADM40001: The application binaries are saved in /optIBM/WebSphere/AppServer/profiles/Dmgr01/estemp514564614workspace/cells/wasv90Cell01/applications/DayTrader EEE/ear/DayTrader EEE.ear.

ADM40051: The application DayTrader EEE is configured in the WebSphere Application Server repository.

SEC30400: Successfully updated the application DayTrader EEE with the appContentID or Security information.

ADM40051: The application DayTrader EEE is configured in the WebSphere Application Server repository.

ADM40051: The application DayTrader EEE is configured in the WebSphere Application Server repository.

ADM40113: Activation plan created successfully.

ADM40111: The cleanup of the temp directory for application DayTrader EEE is complete.

Application DayTrader EEE 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.
- Revert changes before saving or discarding.

To work with installed applications, click the “Manage Applications” link.

[Manage Applications](#)



Step 9: Select the application and action as "Start" and click "Submit Action".

WebSphere software

All Applications

Name	Edition	Edition State	Type	Status	Action
<input checked="" type="checkbox"/> DayTrader_EE5	Base edition	Active	Java 2 Platform, Enterprise Edition	*	<input type="button" value="Start"/>
<input type="checkbox"/> DefaultApplication	Base edition	Active	Java 2 Platform, Enterprise Edition	+	<input type="button" value="Start"/>
<input type="checkbox"/> XOCGIVT	Base edition	Active	Java 2 Platform, Enterprise Edition	+	<input type="button" value="Start"/>

You should see the success message as follows.

WebSphere Integrated Solutions Console - Mozilla Firefox

All Applications

Messages

- Application DayTrader2-EE5 on server App_Server01 and node FusionClouds_dmgrNode01 started successfully. The collection may need to be refreshed to show the current status.
- Application DayTrader2-EE5 on server App_Server02 and node node2Node01 started successfully.
- Application DayTrader2-EE5 started successfully on all of the servers in cluster FC_Cluster.

All Applications

Name	Edition	Edition State	Type	Status	Action
<input type="checkbox"/> DayTrader2-EE5	Base edition	Active	Java 2 Platform, Enterprise Edition	+	<input type="button" value="Start"/>



Step 10: Navigate to “Servers>Server Types>Web servers” and select the web server then click “Generate Plug-in”.

Select	Name	Web server Type	Node	Host Name	Version	Status
<input checked="" type="checkbox"/>	webwasserver	IBM HTTP Server	wasv90Node01	wasv90	ND 9.0.0.0	
Total 1						

Step 11: Click “Propagate Plug-in” to push new configuration to the web server.

Messages

- PLC000051: Plug-in configuration file = /opt/IBM/WebSphere/AppServer/profiles/Dmgr01/config/cells/wasv90Cell01/nodes/wasv90Node01/servers/webwasserver/plugin-cjx.xml
- PLG000521: Plug-in configuration file generation is complete for the Web server.

PLG000521: Propagation of the configuration file is complete for the Web server.

Select	Name	Web server Type	Node	Host Name	Version	Status
<input checked="" type="checkbox"/>	webwasserver	IBM HTTP Server	wasv90Node01	wasv90	ND 9.0.0.0	
Total 1						



Step 11: Check the message and make sure that new plug-in configuration is pushed to the web server.

Select	Name	Web Server Type	Node	Host Name	Version	Status
<input type="checkbox"/>	webserver	IBM HTTP Server	wasv90Node01	wasv90	ND 9.0.0	



Step 11: Navigate to your web server to check the application.
(<https://ihs.fusionclouds.com/daytrader>).

Task 2 is complete!

SUMMARY

WebSphere Application Server supports three different types of applications to be deployed that are enterprise applications, business level applications and assets. You can deploy those applications from your system or you can send the files to the system that hosts deployment manager. Before installing the application, you need to setup the environment according to the needs of application like JMS and JDBC settings. In order to access the application through web server, you have to make sure that you mapped the application modules to the web server, generate and propagate the plug-in configuration to the web server.

REFERENCES

- <http://publib.boulder.ibm.com/infocenter/iadthelp/v6r0/index.jsp?topic=/com.ibm.etools.j2c.ims.doc/tasks/timsj2cwasinstear.html>
- http://pic.dhe.ibm.com/infocenter/clmhelp/v3r0/index.jsp?topic=%2Fcom.ibm.jazz.install.doc%2Ftopics%2Fc_deploying_was.html
- https://www14.software.ibm.com/webapp/download/preconfig.jsp?id=2011-06-08+10%3A34%3A22.216702R&S_TACT=&S_CMP=

INDEX

Asset	306
Business Level Application	306
EAR.....	306
Enterprise Applications.....	306
JAR	306
WAR	306

