

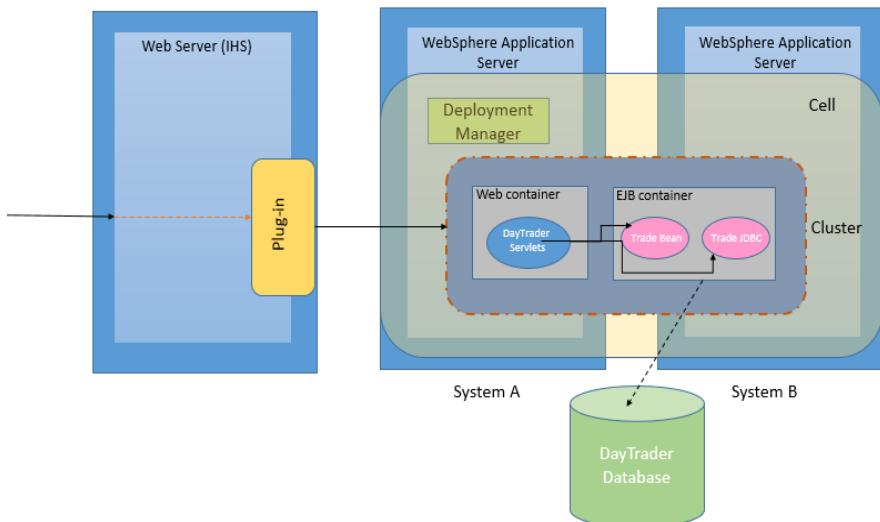
CHAPTER 10: APPLICATION DEPLOYMENT

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 console interface. The left sidebar menu is visible with various navigation options like Welcome, Guided Activities, Servers, Applications, Jobs, Services, Resources, Runtime Operations, and Security. The 'Security' section is expanded, and 'Global security' is selected, which is highlighted with a red box in the screenshot. The main content area displays the 'Cell-easy90Cell01_Profile-Draft1' configuration under 'Global security > JAAS - J2C authentication data'. It provides a brief description: 'Specifies a list of user identities and passwords for Java(TM) 2 connector security to use.' There is a checked checkbox for 'Prefix new alias names with the node name of the cell (for compatibility with earlier releases)'. Below this is an 'Apply' button. The central part of the screen shows a table titled 'Preferences' with columns for 'Select', 'Alias', 'User ID', and 'Description'. A red box highlights the 'New...' button in the top-left corner of this table. The table currently shows 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 '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: *********
- Description:

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: *********
- Description:

Apply | OK | Reset | Cancel |



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

Screenshot of the WebSphere Integrated Solutions Console - Mozilla Firefox window. The URL is <https://dmgr/fusionclouds.com:9043/ibm/console/navigatorCmd.do?csrid=965545255&forwardName=repository.configs>. The left sidebar shows a tree structure with 'Resources' selected. Under 'Resources', 'JDBC' is expanded, and 'JDBC providers' is selected. The main content area shows the 'JDBC providers' configuration page. A red box highlights the 'New...' button in the 'Preferences' section. The status bar at the bottom right shows 'Cell=FusionClouds_dmgrCell01'.

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

Screenshot of the 'Create a new JDBC Provider' wizard. The current step is 'Step 3: Summary'. The configuration fields are as follows:

- Scope:** cells:wasv90Cell01
- Database type:** DB2
- Provider type:** DB2 Universal JDBC Driver Provider
- Implementation type:** XA data source
- Name:** DB2 Universal JDBC Driver Provider (XA)

A red box highlights the 'Name' field. The status bar at the bottom right shows 'Cell=fusionCell01_Profle-Dmgr01'.



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

The screenshot shows the "Create a new JDBC Provider" wizard in progress. The current step is "Step 3: Summary". The left sidebar lists various configuration categories like JMS, JNDI, and Security. The main panel displays the "Enter database class path information" section. It includes fields for "Class path:" containing the value "\$DB2UNIVERSAL_JDBC_DRIVER_PATH/db2jcc.jar", "Native library path:" containing the value "\$DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH", and a note about setting environment variables. A red box highlights the "Next>" button at the bottom of the panel.



Step 6: Click "Finish" to complete.

Screenshot of the WebSphere Integration console showing the 'Create a new JDBC Provider' wizard, Step 3: Summary. The summary table contains the following information:

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 XA 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\$optIBMDB2Drivers
Native path	\$DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH\$
Implementation class name	com.ibm.db2.jcc.DB2XADDataSource

At the bottom of the screen, the 'Finish' button is highlighted with a red box.



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

The screenshot shows the "Data sources" page in the WebSphere Application Server Administration Console. The left sidebar shows "Services" and "Resources" expanded, with "Data sources" selected. The main panel shows a table of existing data sources, including one named "built-in derby-datasource". A red box highlights the "New..." button in the toolbar. The "Scope" dropdown menu is open, showing "Cell:wasv90Cell01" selected. The "Scope help" tooltip is visible, explaining that it specifies the level at which the resource definition is visible. The "Field help" tooltip is also visible.

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

The screenshot shows the "Create a data source" wizard in progress, specifically Step 6: Summary. The "Enter basic data source information" step has been completed, and the "Scope" field is set to "Cell:wasv90Cell01". The "Data source name" field contains "TradeDS" and the "JNDI name" field contains "[JNDI name] [jdbc/TradeDS]". A red box highlights the "Next" button. The "Field help" tooltip is visible.



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

This screenshot shows the "Create a data source" wizard on step 2. The left sidebar lists various resources like JMS, JDBC, and WebSphere variables. The main panel is titled "Selected JDBC provider" and shows the following steps:

- Step 1: Enter basic data source information
- Step 2: Selected JDBC provider (highlighted with a red box)
- Step 3: Enter database specific properties for the data source
- Step 4: Setup security aliases
- Step 5: Summary

The "Selected JDBC provider" section shows a radio button for "Select an existing JDBC provider" which is selected (highlighted with a red box). A dropdown menu next to it shows "DB2 Universal JDBC Driver Provider (XA)". At the bottom, the "Next" button is highlighted with a red box.

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

This screenshot shows the "Create a data source" wizard on step 3. The left sidebar is identical to the previous screenshot. The main panel is titled "Enter database specific properties for the data source" and displays the following properties:

Name	Value
Driver type	4 -
Database name	mywasdb
Server name	wasy99
Port number	59000

A checkbox at the bottom, "Use this data source in container managed persistence (CMP)", is checked. The "Next" button at the bottom is highlighted with a red box.



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

The screenshot shows the "Create a data source" wizard in progress, specifically Step 4: Setup security aliases. The "Authentication alias for XA recovery" dropdown is set to "wasv90cellmanager01/MyWASDataSourceAuthData". The "Container-managed authentication alias" dropdown is set to "wasv90cellmanager01/MyWASDataSourceAuthData". The "Mapping-configuration alias" dropdown is set to "(none)". The "Container-managed authentication alias" dropdown is also set to "(none)". The "Next" button at the bottom is highlighted with a red box.



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

The diagram illustrates a workflow. At the top left, there is a yellow 'UNDER CONSTRUCTION' sign icon. Below it, two rounded rectangular boxes are shown: a red box labeled 'JMS & JDBC' and a green box labeled 'Deploy App.'. A large pink arrow points from the right side of the 'JMS & JDBC' box towards the 'Deploy App.' box. To the right of the arrow, a screenshot of the WebSphere Integration console is displayed. The screenshot shows the 'Create a data source' wizard, Step 5: Summary. It lists various configuration parameters for a data source named 'NotTxTradeDS'. The 'Scope' is set to 'cells:wasv90Cell'. The 'Implementation class name' is 'com.ibm.db2.jcc.DB2XADataSource'. The 'Driver type' is '4'. The 'Database name' is 'mywasdb'. The 'Server name' is 'wasv90'. The 'Port number' is '50900'. There is also a section titled 'Use this data source in container managed persistence (CMP)' which includes fields for 'Authentication alias for XA recovery' (set to 'wasv90CellManager01.MyWASDataSourceAuthData') and 'Component-managed authentication alias' (set to 'wasv90CellManager01.MyWASDataSourceAuthData'). At the bottom of the wizard, there are 'Previous', 'Finish', and 'Cancel' buttons. The 'Finish' button is highlighted with a red border.



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

Screenshot of the WebSphere Integrate software interface. The left sidebar shows navigation options 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, and Buses. The main panel displays a list titled "Buses" with one item: "Createbus(displayName)". Below the list is a toolbar with "New...", "Delete", and other icons. A red box highlights the "New..." button. A tooltip for "Field help" is visible on the right side of the screen.

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

Screenshot of the "Create a new Service Integration Bus" wizard. It is Step 1: Create a new bus. The form has fields for "Enter the name for your new bus" (containing "WAS_Cluster") and "Bus security" (with a checked checkbox). A red box highlights the "Next" button at the bottom of the wizard. A tooltip for "Field help" is visible on the right side of the screen.



Step 16: Click “Next” to continue.

This screenshot shows the 'BusSecurityWizard.displayName' configuration page in the WebSphere Integration console. The left sidebar lists various management categories. The main panel displays the 'Configure security for the bus' wizard, specifically the 'Introduction' step. The 'Next' button at the bottom of the wizard panel is highlighted with a red box.

Step 17: Click “Next” to continue.

This screenshot shows the 'BusSecurityWizard.displayName' configuration page in the WebSphere Integration console, continuing from the previous step. The main panel now displays the 'Specify transport level security' step of the wizard. The 'Specify transport level security' section is highlighted with a blue background. The 'Next' button at the bottom of the wizard panel is highlighted with a red box.



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

The screenshot shows the 'Configure bus security' wizard in the WebSphere Integrator console. The left sidebar lists various integration components like JMS & JDBC and Deploy App. The main window is titled 'Configure bus security' and shows the 'Step 13: Select the security domain for the bus' step. It provides instructions for configuring a bus with members from Version 7 and above. Three radio buttons are available: '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 'Create new security domain' link. At the bottom of the wizard, there are 'Previous', 'Next', and 'Cancel' buttons, with 'Next' being highlighted with a red box.



Step 19: Click "Next" to continue.

Screenshot of the WebSphere software interface. The left sidebar shows a tree view of management categories. The main window displays the "Configure security for the bus" wizard, specifically Step 14: Confirm the enablement of security. It shows a summary of selected options: 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), and Bus security domain (Inheriting the cell level domain). The "Next" button is highlighted with a red box.

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

Screenshot of the WebSphere software interface. The left sidebar shows a tree view of management categories. The main window displays the "Create a new Service Integration Bus" wizard, specifically Step 2: Confirm create of new bus. It shows a summary of actions: New bus "WAS Cluster" will be created with bus security setting "Enabled". The "Finish" button is highlighted with a red box.



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

Step 22: Click “Add”.



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

The screenshot shows the 'WebSphere software' interface with the 'View: All tasks' menu open. The left sidebar lists various system components like Applications, Jobs, Services, and Resources. The main window displays the 'Add bus member' wizard, Step 1: Select server, cluster or WebSphere MQ server. It asks to choose a server, cluster, or WebSphere MQ server to add to the bus. Three options are shown: 'Server' (selected), 'Cluster' (highlighted with a red box and 'WAS_CLUSTER' chosen), 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 the 'View: All tasks' menu open. The left sidebar lists various system components. The main window displays the 'Select Cluster Topology Pattern' wizard, Step 1: Select server, cluster or WebSphere MQ server. It asks to select a predefined messaging engine policy to apply to the selected cluster. 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 unchecked (highlighted with a red box). The 'Next' button at the bottom is also highlighted with a red box.



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

WebSphere Integrat... Close page

Select message store type

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

Select the type of message store

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

Data store

Step 1: Select server, cluster or WebSphere MQ settings

Step 1.1: Messaging engine policy assistance settings

Step 1.1.1: Select the type of message store

(The next step will be displayed depending on decisions made in the current step)

Step 2: Summary

Previous Next Cancel

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

WebSphere Integrat... Close page

Configure a data store

Configure the properties for a data store

Specify data store properties

Specify the properties for the data store

Data source (JNDI name): jdbc/MEDDataSource

Schemas (comma-separated): IBMSSIB

Authentication alias: wasv90CellManager01/MyWASDataSourceAuthData

Create tables

Restrict long running locks

Step 1: Select server, cluster or WebSphere MQ settings

Step 1.1: Messaging engine policy assistance settings

Step 1.1.1: Select the type of message store

Step 1.1.2: Specify data store properties

Step 2: Summary

Previous Next Cancel



Step 27: Select “Next” to continue.

The screenshot shows the 'Tune application server for messaging performance' step. The 'Tune performance parameters' section is visible. At the bottom of the screen, the 'Previous', 'Next', and 'Cancel' buttons are shown, with the 'Next' button being highlighted by a red box.

Step 28: Click “Finish” to complete.

The screenshot shows the 'Add a new bus member' step. The 'Summary' section contains a list of actions to be performed when selecting 'Finish': 'Adding server cluster "WAS_CLUSTER" as member of bus "WAS_Cluster"', 'Data store settings:', 'Data source name "ibmDb/METradeDS"', 'Schema name "IBMWSSIB"', 'Authentication alias "wasm90CellManager01/MyWASDataSourceAuthData"', 'Create connection pool "ibmDb"', and 'Restrict long running locks "false"'. The 'Performance settings:' section is also visible. At the bottom of the screen, the 'Previous', 'Finish', and 'Cancel' buttons are shown, with the 'Finish' button being highlighted by a red box.



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 left sidebar shows navigation paths like "View All tasks", "Buses", "WAS_Cluster", "Bus members", and "Messaging engines". The main content area displays a table with one row for "WAS_CLUSTER-006-WAS_Cluster". A red box highlights the "Add messaging engine" button in the toolbar above the table. The status bar at the bottom indicates "Total 1".

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

Screenshot of a "Select message store type" dialog box. It contains two tabs: "Step 1: Select the type of message store" and "Select the type of message store". The second tab is active, showing a list with "Data store" selected. A red box highlights the "Data store" option. At the bottom, a red box highlights the "Next" button. The left sidebar of the console is visible on the left.



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

Screenshot of the WebSphere Integrator configuration interface. The left sidebar shows "View: All tasks" and a list of administrative categories. The main panel is titled "Configure a data store" and "Specify data store properties". It shows the following configuration:

- Data source (JNDI name): jdbc/METradeDS
- Schema name: IBMWSSIB
- Authentication alias: wasy90CellManager01/MyWASDataSourceAuthData
- Checkboxes: Create tables (checked), Restrict long running locks (unchecked)

The "Next" button is highlighted with a red box.

Step 32: Click “Finish” to complete.

Screenshot of the WebSphere Integrator configuration interface. The left sidebar shows "View: All tasks" and a list of administrative categories. The main panel is titled "Configure the properties for a data store" and "Summary". It shows the following summary information:

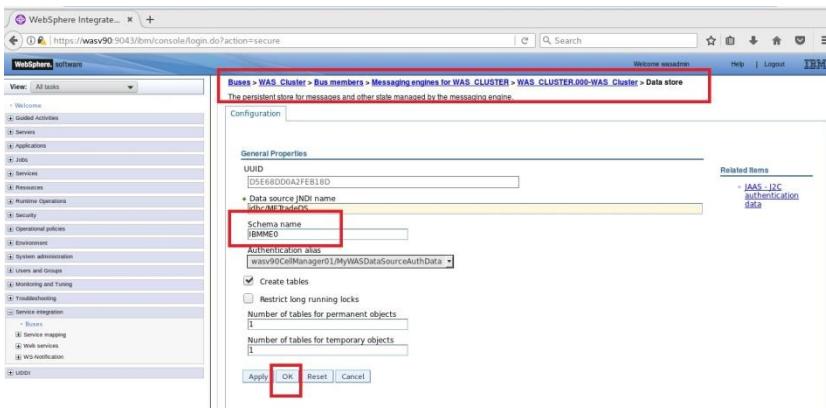
The actions that will be performed when selecting "Finish":

- Data store settings:
 - Data source (JNDI name): "jdbc/METradeDS"
 - Schema name: "IBMWSSIB"
 - Authentication alias: "wasy90CellManager01/MyWASDataSourceAuthData"
 - Create tables: "true"
 - Restrict long running locks: "false"

The "Finish" button is highlighted with a red box.



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 contains two screenshots of the WebSphere Integration console interface. Both screenshots show the left sidebar with categories like Welcome, Guided Activities, Servers, Applications, and System administration. The main area shows the "Core Groups" page for a specific server profile.

Screenshot 1 (Top): Shows the "Core Groups" page. A red box highlights the "Core Groups" link under the "Servers" category in the sidebar. Another red box highlights the "DefaultCoreGroup" entry in the list of core groups. The right panel displays details for the DefaultCoreGroup, including its name, description ("Default Core Group. The default core group cannot be deleted."), and a table showing connected core groups. A red box highlights the "DefaultCoreGroup" entry in the table.

Screenshot 2 (Bottom): Shows the "Core Groups > DefaultCoreGroup" configuration page. A red box highlights the "DefaultCoreGroup" entry in the breadcrumb navigation. The right panel has tabs for Runtime, Configuration, and Operations. Under General Properties, fields include Name (DefaultCoreGroup), Description (Default Core Group. The default core group cannot be deleted.), Number of coordinators (1), and Transport memory size (100 megabytes). Under Additional Properties, there are sections for Core group servers, Discovery and coordination, Policies (which is highlighted with a red box), and Custom properties. Under Related Items, there is a link to Core group bridge settings.



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.

Select	Name	Description	Policy type	Match criteria
<input type="checkbox"/>	Clustered TM Policy	TM One-Of-N Policy	One of N policy	type=WASF_TRANSACTIONS
<input type="checkbox"/>	Default SIBus Policy	SIBus One-Of-N Policy	One of N policy	type=WSAF_SIB
<input type="checkbox"/>	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

Core Groups > DefaultCoreGroup > Policies > New...

Specifies the type of policy that you are creating.

Configuration

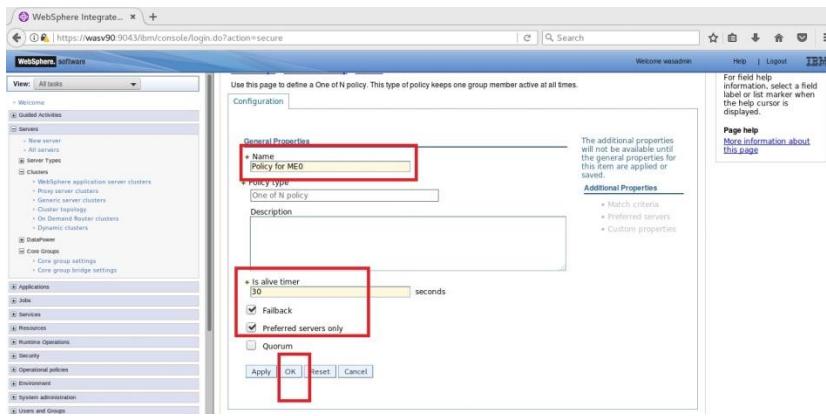
General Properties

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”.





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.

The 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 "Select", "Name", "Type", and "Description". One row is selected, showing "Server" as the name and "Group" as the type. A red box highlights the "New..." button in the top navigation bar.

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

The 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 destinations under "Buses > WAS_Cluster > Destinations". The table has columns "Select", "Identifier", "Bus", "Type", "Description", and "Mediation". Several rows are listed, including "Default Topic Space", "SYSTEM.Exception Destination WAS_CLUSTER.000-WAS_Cluster", and "SYSTEM.Exception Destination WAS_CLUSTER.001-WAS_Cluster". A red box highlights the "New..." button in the top navigation bar.



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

The screenshot shows the 'Create new destination' wizard in the WebSphere Integrate console. On the left is a navigation sidebar with various categories like Servers, Applications, and Security. The main panel is titled 'Create new destination' and has a sub-section 'Create a new destination on this bus.' It asks to 'Select destination type' with options: Queue (selected), Topic space, Alias, and Foreign. Below this 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 'Create new queue' wizard in the WebSphere Integrate console. On the left is a navigation sidebar. The main panel is titled 'Create new queue' and has a sub-section 'Set queue attributes' with a sub-sub-section 'Configure the attributes of your new queue'. It shows an 'Identifier' field containing 'TradeBrokerJSD' and a 'Description' field. Below this is a 'Next' button, which is highlighted with a red box. A 'Field help' tooltip is visible on the right side of the screen.



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

Screenshot of the WebSphere Integrator console showing the 'Create new queue' wizard, Step 2: Assign the queue to a bus member. The 'Bus member' dropdown is set to 'Cluster=WAS_CLUSTER'. The 'Next' button at the bottom is highlighted with a red box.

Step 46: Click “Finish”.

Screenshot of the WebSphere Integrator console showing the 'Create new queue' wizard, Step 3: Confirm queue creation. The summary box indicates 'New queue "TradeBrokerJSD" will be created' and 'Queue points for "TradeBrokerJSD" will be created for bus member "WAS_CLUSTER" of bus "WAS_Cluster"'. The 'Finish' button at the bottom is highlighted with a red box.



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.

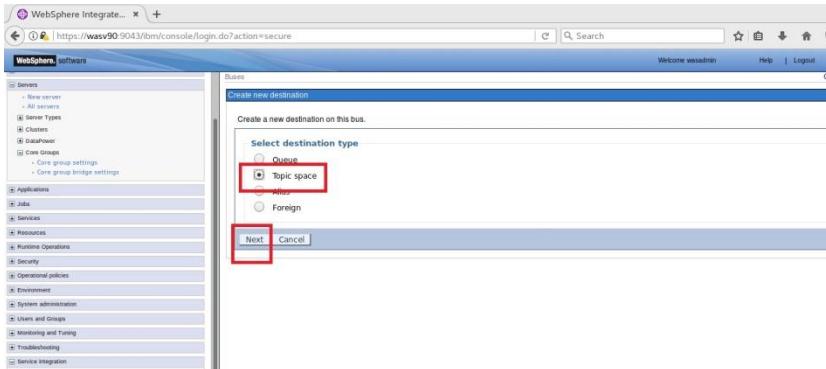
The screenshot shows the WebSphere Integration console interface. On the left, there is a navigation tree with categories like Servers, Applications, Services, Resources, and System Administration. In the main panel, there is a form for configuring a destination. The "Type" field is set to "Queue". The "Quality of Service" section is highlighted with a red box. It contains the following settings:

- Enable producers to override default reliability
- Default reliability: Express nonpersistent
- Maximum reliability: Assured persistent

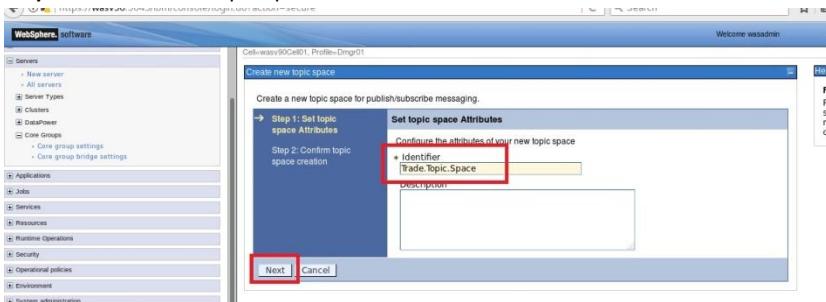
Below this, there are sections for "Mediation", "Related Items", "Default priority", and "Exception destination".



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

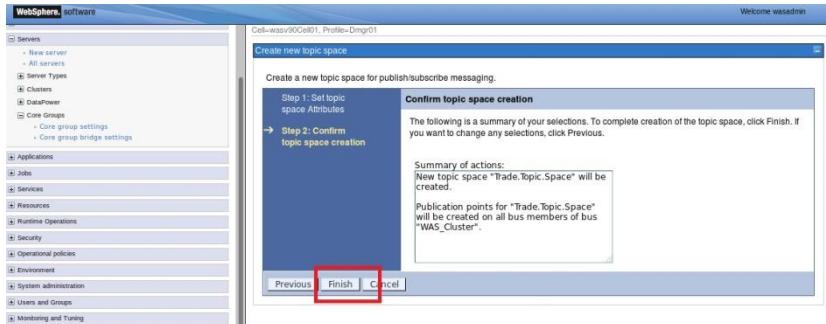


Step 49: Use “Trade.Topic.Space” as identifier and click “Next”.

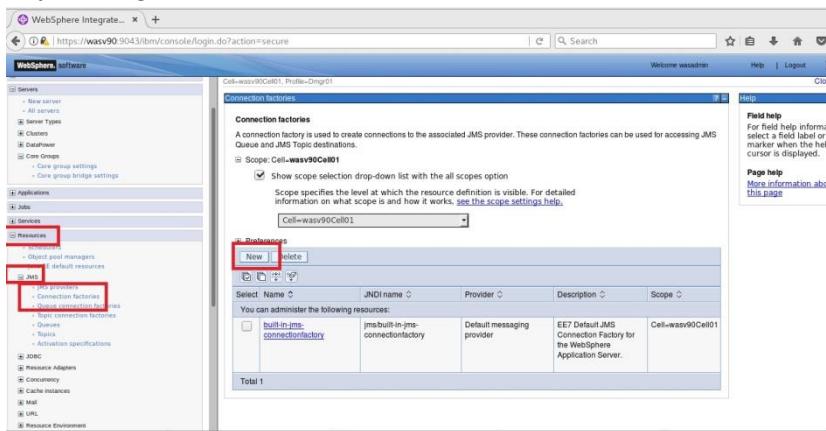




Step 50: Click "Finish" to complete.



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





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

Screenshot of the WebSphere Integration console showing the "Connection factories" configuration page. The left sidebar shows the navigation tree with "JMS" selected under "Resources". The main panel shows a list of connection factories. A specific connection factory is selected, and its properties are displayed in the center. The "Provider" dropdown is set to "Default messaging provider", which is highlighted with a red box. At the bottom of the dialog, the "OK" button is also highlighted with a red box.

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

Screenshot of the WebSphere Integration console showing the "General Properties" configuration page for a JMS connection factory. The left sidebar shows the navigation tree with "JMS" selected under "Resources". The main panel displays various properties for the connection factory. The "Name" field is set to "TradeBrokerQCF" and the "JNDI name" field is set to "jms/TradeBrokerQCF", both highlighted with red boxes. In the "Connection" section, the "Bus name" dropdown is set to "WAS_Cluster", which is also highlighted with a red box.



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

The screenshot shows the WebSphere Integration console interface. On the left, there is a navigation tree with categories like Servers, Applications, Services, and Resources. Under Resources, the JMS section is expanded, showing options like JMS providers, Connection factories, and Topics. In the main panel, there is a configuration dialog for a JMS connection factory. The "Mapping-configuration alias" dropdown menu is open, and the option "DefaultPrincipalMapping" is selected and highlighted with a red box. At the bottom right of the dialog, the "OK" button is also highlighted with a red box. Other visible fields include "Authentication alias for XA recovery" (set to "None") and "Activation specifications" (set to "(none)").



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

Screenshot of the WebSphere Integration console showing the 'Topic connection factories' page. The left sidebar shows 'Resources' expanded, with 'JMS' selected. The main panel displays the 'Topic connection factories' configuration with a 'Producers' section containing a 'New' button. A red box highlights the 'New' button.

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

Screenshot of the 'Topic connection factories > Select JMS resource provider' dialog. It shows a list of providers: 'Default messaging provider' (selected) and 'WebSphere MQ messaging provider'. A red box highlights the 'OK' button at the bottom of the dialog.



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

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



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

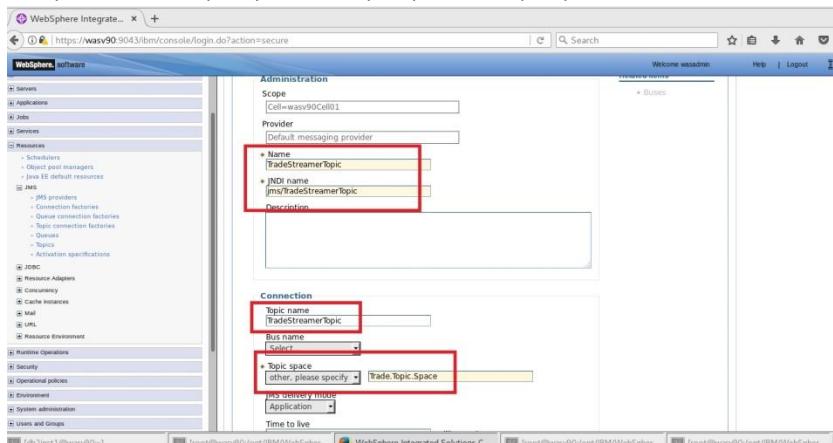
Screenshot of the WebSphere Integration console showing the 'Select JMS resource provider' dialog. The 'Scope' field is set to 'Cell:wasv90Cell01'. The 'Provider' dropdown shows 'Default messaging provider' selected. Below it, 'WebSphere MQ messaging provider' is listed. The 'OK' button at the bottom is highlighted with a red box.

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

Screenshot of the WebSphere Integration console showing the 'Create Queue' dialog. The 'Scope' field is set to 'Cell:wasv90Cell01'. The 'Provider' dropdown shows 'Default messaging provider'. Under 'Name', 'TradeBrokerQueue' is selected. Under 'JNDI name', 'jms/TradeBrokerQueue' is entered. Under 'Connection', 'Bus name' is set to 'Local'. Under 'Delivery mode', 'Queue name' is set to 'other, please specify' with 'TradeBrokerJSD' entered. The 'OK' button at the bottom is highlighted with a red box.



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.

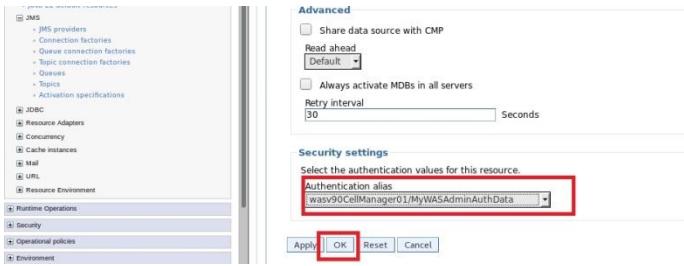
The screenshot shows the WebSphere Integrations Console interface. On the left, there is a navigation tree with various service categories like Servers, Applications, Services, and Resources. The main panel is titled "Provider" and shows configuration for the "Default messaging provider". The "Name" field is set to "TradeBrokerMDB" and the "JNDI name" field is set to "jms/TradeBrokerMDB". Below this, the "Destination" section is configured with "Destination type" set to "Queue", "Destination lookup" set to "TradeBrokerQueue", and "Connection Factory lookup" set to "TradeBrokerQueue". The "Message selector" section has "Bus name" set to "WAS_Cluster" and "Acknowledge mode" set to "Auto-acknowledge".

Step 64: Use “Subscription Durability” as follows.

The screenshot shows the "Subscription Durability" configuration screen. The "Subscription durability" dropdown is set to "Non-durable". The "Subscription name" field is set to "TradeBrokerMDB". The "Client identifier" field is set to "TradeBrokerMDB". The "Subscription home" dropdown is set to "WAS_CLUSTER.000-WAS_Cluster". Below this, there is a section for "Pass message payload by reference" with a checkbox that is unchecked. The checkbox text reads: "Applications using this Activation Specification to receive messages: do not modify the data object obtained from a JMS Object Message. The data object is treated as read only. Read the help before selecting this option."

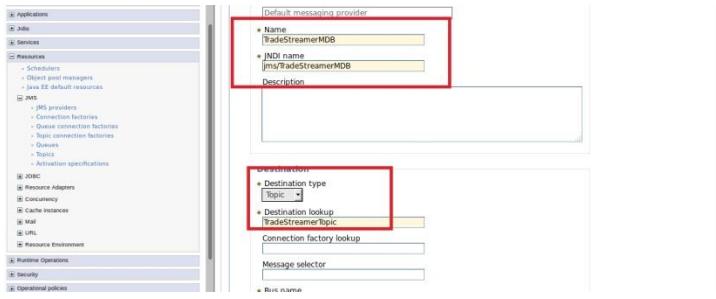


Step 65: Use “MyWASAdminAuthData” as authentication alias and click “OK”.





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”.

The top screenshot shows the 'Subscription Durability' configuration dialog. It includes fields for 'Subscription durability' (set to 'NonDurable'), 'Subscription name' ('TradeStreamerMDB'), 'Client identifier' ('TradeStreamerMDB'), and 'Subscription home' ('WAS_CLUSTER.000-WAS_Cluster'). The bottom screenshot shows the 'Pass message payload by reference' configuration dialog, which includes sections for 'Client identifier' ('TradeStreamerMDB'), 'Subscription home' ('WAS_CLUSTER.000-WAS_Cluster'), 'Share subscriptions' ('In cluster'), and various checkboxes related to message processing and activation specifications.

Task 1 is complete!



Task 2: Deploy a sample application

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

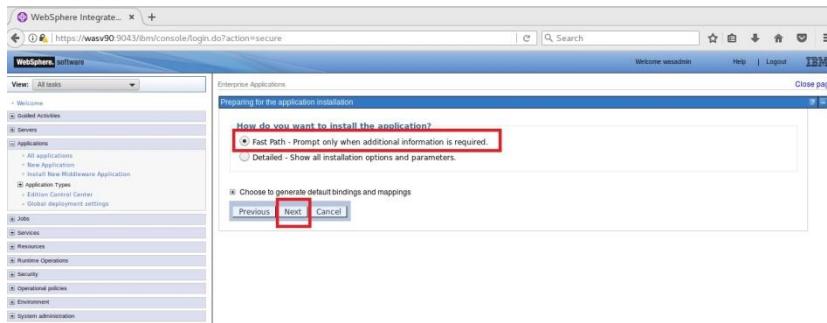
The screenshot shows the WebSphere Integration console interface. On the left, there's a navigation sidebar with various links like 'Welcome', 'Guided Activities', 'Services', 'Applications', 'Jobs', 'Services', 'Resources', etc. The 'Applications' link is currently selected. On the right, a 'New Application' dialog box is open. It has a title 'New Application' and a sub-section 'Install a New Application'. Under this section, there are three options: 'New Enterprise Application' (which is highlighted with a red box), 'New Business Level Application', and 'New Asset'. Below the dialog, the main console interface shows a tree view of 'Enterprise Applications' and other system components.

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

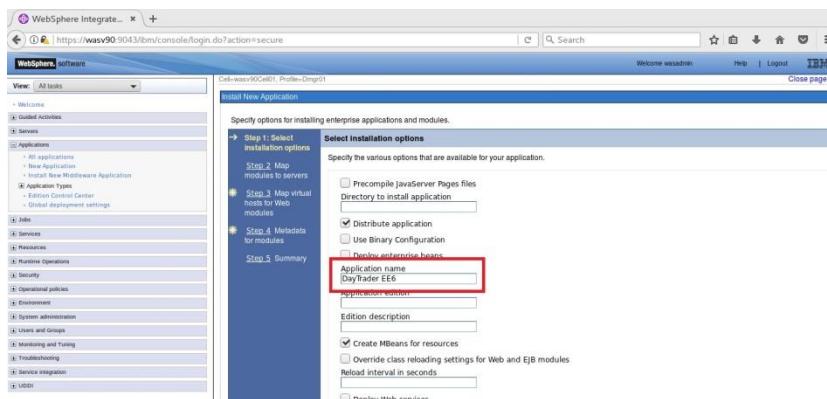
This screenshot shows the 'Preparing for the application installation' step of the deployment wizard. The left sidebar is identical to the previous screenshot. The main area has a title 'Preparing for the application installation' and a sub-section 'Specify the EAR, WAR, JAR, or SAR module to upload and install'. It shows a 'Path to the new application' field with two radio button options: 'Local file system' (unchecked) and 'Remote file system' (checked). The 'Remote file system' path is set to 'file:///root/software/LinuxApp/DayTrader3/install/DayTrader3-EE6/daytrader'. At the bottom are 'Next' and 'Cancel' buttons, with 'Next' also highlighted with a red box. A 'Help' panel 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	web.war WEB-INF/web.xml	WebSphere cell=wasv90Cell01,node=wasv90Node01,server=server1
<input type="checkbox"/>	Rest.war	Rest.war IWEB-INF/web.xml	WebSphere cell=wasv90Cell01,node=wasv90Node01,server=server1

Buttons at the bottom: Previous, Next (highlighted with a red box), Cancel.



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.

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 workspace Help | Logout

View: All tasks

Applications

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

Jobs

Services

Processes

Runtime Operations

Security

Operational policies

Environment

System administration

Users and Groups

Monitoring and Testing

Troubleshooting

Service Integrator

UDI

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

Search

ADM40113W: Resource assignment of file /usr/local/tomcat/tomcat-7.0.52/bin/libexec/java/lib/jar/resource, with file name pcom/rabbitmq/resource is not found within scope of module play/rabbitmq with identifier was:WEB-INF-Avuls.wim deployed to target WebSphereCell/wasv90Cell02/cluster/IWAS_CLUSTER.

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

ADM40081: Application and module versions are validated with versions of deployment targets.

ADM40081: The application DayTrader EEE is configured in the WebSphere Application Server repository.

ADM40081: The application DayTrader EEE is configured in the WebSphere Application Server repository.

ADM40081: The bootstrap address for client module is configured in the WebSphere Application Server repository.

ADM40081: The library references for the installed optional package are created.

ADM40081: The application DayTrader EEE is configured in the WebSphere Application Server repository.

ADM40081: The application binaries are saved in /opt/BMW/WebSphere/AppServer/profiles/Dmgr01/estemp514564614/workspace/cells/wasv90Cell01/applications/DayTrader EEE/ear/DayTrader EEE.ear.

ADM40081: 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.

ADM40081: The application DayTrader EEE is configured in the WebSphere Application Server repository.

ADM40081: 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.

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.
- Review 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".

Screenshot of the WebSphere Integrated Solutions Console showing the 'All Applications' page. The 'DayTrader_EE5' application is selected, and the 'Start' button is highlighted with a red box.

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.

Screenshot of the WebSphere Integrated Solutions Console showing the 'Messages' section with three success 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. The collection may need to be refreshed to show the current status.
- Application DayTrader2-EE5 started successfully on all of the servers in cluster FC_Cluster.

The 'All Applications' table below also has a red box around the 'Start' button for the 'DayTrader_EE5' row.

Name	Edition	Edition State	Type	Status	Action
<input type="checkbox"/> DayTrader_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.



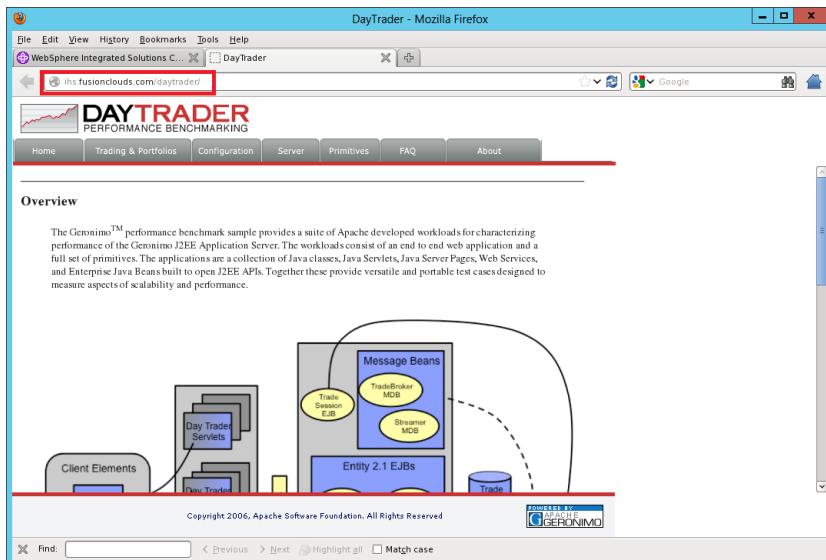
Step 11: Check the message and make sure that new plug-in configuration is pushed to the web server.

Screenshot of the WebSphere Admin Console interface. The URL is https://wasv90:9043/bm/console/login.do?action=secure. The title bar shows 'WebSphere software' and the sub-title 'Cell=wasv90Cell01, Profile=Dmgr01'. The main menu on the left includes 'Welcome', 'Guided Activities', 'Servers' (selected), 'Service Types', 'Applications', 'Jobs', 'Services', and 'Resources'. The 'Servers' section is expanded, showing 'Web servers' (selected) and a list of server types: WebSphere application servers, WebSphere enterprise servers, WebSphere proxy servers, On-Demand Routers, WebSphere MQ servers, WebSphere Application Server Community Edition servers, Diagnostic servers, WebSphere MQ servers, JBoss servers, Apache Tomcat servers, BEA WebLogic servers, IBM HTTP Server, External WebSphere Application Servers, Apache servers, and Custom HTTP servers. The 'Web servers' section on the right lists 'Web servers'. It shows a table with one row for 'wasv90Node01.webserver'. The table columns are: Select, Name, Web server type, Node, Host Name, Version, and Status. The status for 'wasv90Node01.webserver' is 'ND 9.0.0.0'. A red box highlights the 'Messages' section, which contains three entries: PLGC0062i, PLGC0048i, and PLGC0074i. The right sidebar includes links for 'Help', 'Page help', 'More information about this page', and 'Command Assistance'.

Select	Name	Web server type	Node	Host Name	Version	Status
<input type="checkbox"/>	wasv90Node01.webserver	IBM HTTP Server	wasv90Node01	wasv90	ND 9.0.0.0	



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

