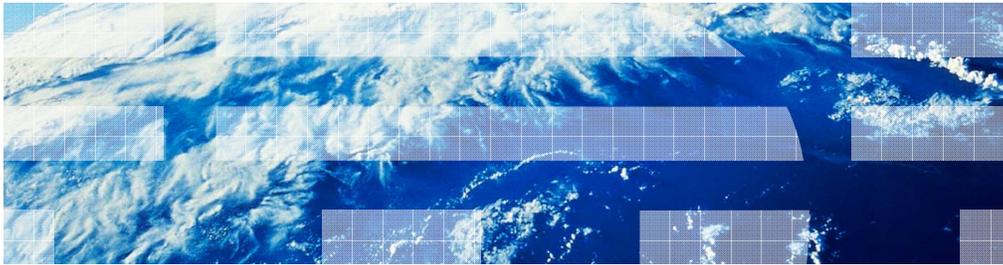


# IBM WebSphere Application Server V8

Application deployment by adding properties file to a monitored directory



An enterprise application can be installed on an application server or cluster by dragging or copying an ear file, war file, jar file or sar file to a monitored directory; or it can be installed or updated using a properties file on a server or cluster. The deployed application can also be uninstalled using the application properties file. This presentation provides an overview of using an application properties file to the monitored directory to install, uninstall or update an enterprise application in WebSphere Application Server version 8.

---

## Table of contents

- Overview
- Deploy application
- Uninstall application
- Update application
- Problem determination
- Summary
- References

This presentation covers an overview to use properties file to the monitored directory to deploy, uninstall or update an application with various install options. It also covers details on problem determination and references of Information Center.

## Overview

- Use properties file based configuration
  - Introduced in V7
  - Provides a group of administrative commands
  - Manage system configuration
  - Troubleshoot configuration issues
  - Replicate configuration properties across profiles, nodes, servers or applications
  - Use properties file on monitored directory deployment to deploy applications

The properties file based configuration feature has been introduced in version 7 and it has provided a group of administrative commands to manage system configuration using properties files. A user can use the properties file based configuration commands to copy configuration properties from one environment to another. He can also use the properties file to troubleshoot configuration issues and apply one set of configuration properties across multiple profiles, nodes, cells, servers or applications. The details of the properties file based configuration commands are documented in version 7 Information Center.

The Monitored Directory deployment has extended to use the properties file to install, uninstall or update an application in version 8.

## Deploy applications using properties files

- Steps to use properties file to deploy applications
- Properties file examples

A user can deploy an EAR, JAR, WAR, or SAR file to an application server or cluster by dragging or copying an application properties file to a monitoredDeployableApps under the deploymentProperties monitored directory. The monitored directory deployment service scans the directory for any new properties files every 5 seconds. After finding a new properties file in a monitored directory, the product automatically runs the wsadmin the properties file based configuration applyConfigProperties command to install the application or module on the application server or cluster, and starts the application or module.

There is no need to start wsadmin or enter any commands to deploy the application or module. Simply add a properties file to a monitored directory and the product will run the applyConfigProperties command for you.

User can also use a properties file to update or delete a deployed application or module. The server or cluster must be running so that the product can detect changes to files in its monitored directory.

It can use a graphical file browser to drag or copy the properties file or use the operating system commands to copy a file into a monitoredDeployableApps deploymentProperties monitored directory.

## Steps to use properties file to deploy applications

- Create a properties file that defines deployment options
  - Use Properties File Based Configuration command to create an application properties file
    - Extract application properties to a file with V7 output format

```
AdminTask.extractConfigProperties(['-propertiesFileName myApp.props -configData
Deployment=MyApplication]')
```

- Extract application properties to a file with simple output format

```
AdminTask.extractConfigProperties(['-propertiesFileName myApp.props -configData
Deployment=MyApplication -options [[SimpleOutputFormat true]]']')
```

- Ensure that the targeted server or cluster member is running
- Ensure that monitored directory deployment is enabled
- Copy the properties file to MonitoredDeployableApps/deploymentProperties directory

A user can use the properties file based configuration command to extract application properties to an application properties file. After a properties file is extracted, edit application properties and copy the properties file to the MonitoredDeployableApps under deploymentProperties directory. The Monitored directory deployment service will run applyConfigProperties command to apply the application properties change to install, uninstall or update an application for you.

The step one shows how to extract application properties to a file using the properties file based configuration extractConfigProperties command.

By default, the extractConfigProperties command produces output that displays all columns including all hidden and non-hidden columns of install tasks and task data values in separate rows. You can also extract application properties using simple output format to only display non-hidden columns of install task data in columnName=value pairs.

You can also use the properties file examples documented in the information center to create a new application properties file to deploy, uninstall or update an application.

The monitored directory deployment service will start the application after it is deployed or update, so you need to ensure the application server or cluster member on which you want to install the enterprise application files is running.

The step 3 is to ensure the monitored directory deployment is enabled. You can look at the information center on setting monitored directory deployment values section.

The last step is to copy the application properties file to the MonitoredDeployableApps directory. You can use a file browser to drag the properties file to the monitored directory or use the operating system command to copy the properties file to the directory

## Deploy an enterprise application

```
# Header
#
ResourceType=Application
ImplementingResourceType=Application

# Properties
Name={!{applicationName}}
TargetServer={!{serverName}}
TargetNode={!{nodeName}}
EarFileLocation={!{earFile}}

EnvironmentVariablesSection
#
#
#Environment Variables
cellName=cellName01
nodeName=nodeName01
serverName=server1
earFile=c:/WebSphere/AppServer/installableApps/DefaultApplication.ear
applicationName=app1
```

This presentation provides a sample application properties file for use in the monitored directories to deploy an enterprise application. It is slightly different from the properties file extracted from the properties file based configuration `extractConfigProperties` command. It has only contained few required application properties to deploy an application.

It does not need to specify statements such as `CreateDeleteCommandProperties=true` in the header.

You only specify application resource type and implementing resource type as “Application” in the properties file. If the properties file contains a non-application resource type such as `ResourceType=Server` or `ImplementingResourceType=Server`, the monitored directory deployment service will return an error message and will not perform the operation on the resource type.

The “Name” is the name of application to install.

The “TargetServer” is the name of server on which application to be installed

The “TargetNode” is the name of node on which application to install.

The “EarFileLocation” is the location of the enterprise archive (EAR file). Provide a fully qualified path name.

You can specify “TargetCluster” if you want to deploy an application on a cluster.

You can specify the application properties such as Name, targetServer, TargetNode or EarFileLocation in the “Properties” section directly or you can specify them in the EnvironmentVariableSection section and the properties file based configuration tool will substitute the variables for you.

The monitored directory deployment service will start the application after it is deployed so you need to ensure the targeted application server and node agent are running. An error message is returned if the application server or node agent is not running.

## Uninstall an enterprise application

```
#  
# Header  
#  
ResourceType=Application  
ImplementingResourceType=Application  
DELETE=true  
  
# Properties  
Name=app1
```

This presentation provides an example properties file to uninstall a deployed application. It does not need to specify DELETE=true statement in the header and application name in the Properties section to uninstall a deployed application.

## Update an enterprise application

```
# Header
#
ResourceType=Application
ImplementingResourceType=Application

# Properties
Name={!applicationName}
Update=true
operationType=update
contentType=app
contentFile={!contentFile}

EnvironmentVariablesSection
#
#
#Environment Variables
applicationName=app1
contentFile=c:/WebSphere/AppServer/installableApps/DynaCacheEsi.ear
```

The presentation provides a properties file example to update an entire deployed application.

Specify `Update=true`, `operationType=update`, and `contentType=app` in the Properties section. The example replaces the `app1` with the ear file `DynaCacheEsi.ear`. The `useDefaultBindings=true` property instructs the product to generate default bindings for the application.

When the full application is updated, the old application is uninstalled and the new application is installed. After new application is installed, the configuration changes are saved and subsequently synchronized. The application files are expanded on the node where application will run. If the application is running on the node while it is updated, then the application is stopped, the application files are updated, and application is started.

## Add a single file to a deployed application

```
# Header
#
ResourceType=Application
ImplementingResourceType=Application

# Properties
Name=!{applicationName}
Update=true
operationType=add
contentType=file
contentURI=!{contentURI}
contentFile=!{contentFile}

EnvironmentVariablesSection
#
#
#Environment Variables
applicationName=app1
contentFile=c:/temp/addMe.jsp
contentURI=test.war/com/ibm/addMe.jsp
```

This presentation provides an example to add a single file to a deployed application.

Edit the application properties file so that it specifies the file to add or update or delete.

Edit the properties of an Application instance. Specify Update=true, an operation type such as operationType=add or update or delete, and contentType=file and contentURI in the Properties section.

The example adds a single file “addMe.jsp” to a deployed application named app1. You can change operationType=update to update a single file in a deployed application or you can change operationType=delete to delete a file in a deployed application.

## Remove a module file from a deployed application

```
# Header
#
ResourceType=Application
ImplementingResourceType=Application

# Properties
Name=!{applicationName}
Update=true
operationType=delete
contentType=modulefile
contentURI=!{contentURI}
contentFile=!{contentFile}

EnvironmentVariablesSection
#
#Environment Variables
applicationName=app1
contentFile=c:/temp/Increment.jar
contentURI=Increment.jar
```

This presentation provides an example to update a module file in a deployed application.

Edit the application properties file so that it specifies the Java EE module to add or change.

Edit the properties of an Application instance. Specify Update=true, an operation type such as operationType=add, update or delete, and contentType=moduleFile and contentURI in the Properties section.

The example updates the Increment.jar file to a deployed application named app1. You can add or delete a Java EE module in a deployed application by changing operationType=add or operationType=delete.

## Update a partial application

```
# Header
#
ResourceType=Application
ImplementingResourceType=Application

# Properties
Name={!applicationName}
Update=true
operationType=update
contentType=partialapp
contentFile={!contentFile}

EnvironmentVariablesSection
#
#
#Environment Variables
applicationName=app1
contentFile=c:/temp/MyApp/myAppPartial.zip
```

This presentation provides an example to update multiple files of an installed application by uploading a compressed file.

Depending on the contents of the compressed file, you can replace files in, add new files to, or delete files from the installed application. Each entry in the compressed file is treated as a single file and the path of the file from the root of the compressed file is treated as the relative path of the file in the installed application.

To replace a file, a file in the compressed file must have the same relative path as the file to be updated in the installed application.

To add a new file to the installed application, a file in the compressed file must have a different relative path than the files in the installed application.

The relative path of a file in the installed application is formed by concatenation of the relative path of the module, if the file is inside a module, and the relative path of the file from the root of the module separated by a forward slash (/).

To remove a file from the installed application, specify metadata in the compressed file using a file named META-INF/ibm-partialapp-delete.props at any archive scope. The ibm-partialapp-delete.props file must be an ASCII file that lists files to be deleted in that archive with one entry for each line. The entry can contain a string pattern such as a regular expression that identifies multiple files. The file paths for the files to be deleted must be relative to the archive path that has the META-INF/ibm-partialapp-delete.props file.

You can look at the “Replace, add or delete multiple files” section in the information center for more information about the metadata .props file to include in compressed files.

The example properties use the myAppPartial.zip compressed file to update a deployed application named app1.

## Deploy an application with V7 output format

```
# Header
ResourceType=Application
ImplementingResourceType=Application

# Properties
Name={!applicationName}
EarFileLocation={!earFile}

# SubSection 1.0.2 # MapModulesToServer Section. taskName # and row0 should not be edited. row0 contains column names for the task.
#
ResourceType=Application
ImplementingResourceType=Application
ResourceId=Deployment={!applicationName}

# Properties
#
taskName=MapModulesToServers
row1={DynaCacheEsi DynaCacheEsi.war,WEB-INF/web.xml WebSphere:cell={!cellName},node={!nodeName},server={!serverName} 23
moduletype.web "Web Module"}
mutables={false false true false false false} #readonly
row0={module uri server ModuleVersion moduletype moduletypeDisplay} #readonly
```

The example properties file installs an enterprise application with the MapModulesToServers install task option.

The properties file uses the version 7 output format to display all columns, including hidden and non-hidden columns of install task and task data values in separate rows. The row1 shows all modules data and the mutables row shows which columns can edit (true) and which cannot edit (false). The row0 shows the column names.

You can only modify the column that are mutable (true) such as WebSphere:cell={cellName},node={!nodeName},server={!serverName}. The EnvironmentVariablesSection is missing in this example due to the space limitation.

## Deploy an application using simple output format

```
# Properties
Name={applicationName}
EarFileLocation={earFile}

# SubSection 1.0.2 # MapModulesToServers Section. taskName and lines marked as "#readonly" should not be edited.
#
ResourceType=Application
ImplementingResourceType=Application
ResourceId=Deployment={applicationName}

# Properties
#
taskName=MapModulesToServers
row1={ module="Default Web Application" #readonly
      uri=DefaultWebApplication.war,WEB-INF/web.xml #readonly
      server=WebSphere:cell={cellName},node={nodeName},server={serverName} }
row0={ module="Increment EJB module" #readonly
      uri=Increment.jar,META-INF/ejb-jar.xml #readonly
      server=WebSphere:cell={cellName},node={nodeName},server={serverName} }
```

The example properties install an enterprise application with the MapModulesToServer task option.

The example shows application properties that have been extracted with the SimpleOutputFormat option. With this option, the properties file displays non-hidden columns of application properties in columnName=value pairs. Hidden columns of application properties are not included in the output. Extracting application properties files with the SimpleOutputFormat option produces a format that is easier to read and edit. The #readonly column cannot be edited.

## Problem determination

- Look at SystemOut.log
  - Search CWLDDxxxx messages for troubleshooting
- Properties file related problems
  - Properties file validation is enabled
    - Syntax error
    - Usage error
  - Report is generated and stored in  
\$WAS\_HOME/profiles/<profilename>/wstemp/<dragdropSession>/<prop>\_report
    - applyConfigProperties command fail
- Enable server traces
  - com.ibm.ws.management.dragdrop.\*=all:com.ibm.ws.management.commands.properties.\*=all:com.ibm.ws.management.wasresources.\*=all

If application is not deployed, uninstalled or updated successfully, first look at the SystemOut.log and read the messages starts with CWLDD key to figure out the problem. If the message indicates application failed to start, ensure application server and its node agent are running. If the message indicates some properties file validation error, check the generated report file located on \$WAS\_HOME/profiles/<profilename>/wstemp/<dragdropSession> directory.

If the message reports some properties file validation error such as the properties file syntax error or user error, fix the properties file error and copy the properties file to the monitored directory again.

If the message reports some error occurred at the applyConfigProperties command execution such as application installation failed, enable server traces for troubleshooting.

## Summary

- Install, update and uninstall an application by adding properties file to a monitored directory without the administrative console or wsadmin
- Support application deployment with various install options specify on the properties file
- Support various types of application update: single file, module file, full application and partial application update
- Support multiple application operations performed on same properties file such as
  - Install a new application, update the application
  - Update a file, update a module of an existing application
  - Uninstall application, install new application, update the application
- It is supported on
  - Standalone Application Servers
  - Network Deployment servers and clusters
  - Distributed operating systems
  - z/OS

The monitored directory deployment is a simple and efficient way to deploy applications without the administrative console or wsadmin. Drag or copy an application properties file to a monitored directory to install enterprise application files on a server or cluster, update deployed applications or modules, or uninstall deployed applications or modules. It currently supports the deployment of EARs, JARs, WARs, and SARs on all the distributed operating systems and z/OS. It also supports various types of application update such as updating a single file of a deployed application or updating entire application, or updating a module file or partial application.

It also allows multiple application operations to be performed on same properties file, for example, user can install a new application then update a module file on the new deployed application, or update a deployed application by deleting a module file, then adding a new module file to the deployed application, or uninstall a deployed application, reinstall it and update one of modules, and so on.

Finally, monitored directory deployment is supported on stand-alone WebSphere Application Servers and Network Deployment servers and clusters.

## References

- Installing enterprise application files by adding them to a monitored directory  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp?topic=/com.ibm.websphere.zseries.doc/info/zseries/ae/trun\\_app\\_install\\_dragdrop.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp?topic=/com.ibm.websphere.zseries.doc/info/zseries/ae/trun_app_install_dragdrop.html)
- Uninstalling enterprise application files by dragging them from a monitored directory  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp?topic=/com.ibm.websphere.zseries.doc/info/zseries/ae/trun\\_app\\_uninst\\_dragdrop.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp?topic=/com.ibm.websphere.zseries.doc/info/zseries/ae/trun_app_uninst_dragdrop.html)
- Installing enterprise application files by adding properties files to a monitored directory  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp?topic=/com.ibm.websphere.zseries.doc/info/zseries/ae/trun\\_app\\_install\\_dragdrop\\_prop.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp?topic=/com.ibm.websphere.zseries.doc/info/zseries/ae/trun_app_install_dragdrop_prop.html)
- Using properties file to manage system configuration  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.base.iseries.doc/info/series/ae/txml\\_property\\_configuration.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.base.iseries.doc/info/series/ae/txml_property_configuration.html)

This slide contains links to useful information.



## Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send email feedback:

[mailto:iea@us.ibm.com?subject=Feedback\\_about\\_WASv8\\_AppDeploy.ppt](mailto:iea@us.ibm.com?subject=Feedback_about_WASv8_AppDeploy.ppt)

This module is also available in PDF format at: [../WASv8\\_AppDeploy.pdf](#)

You can help improve the quality of IBM Education Assistant content by providing feedback.



## Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, WebSphere, and z/OS are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the web at "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at <http://www.ibm.com/legal/copytrade.shtml>

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY.  
Other company, product, or service names may be trademarks or service marks of others.

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.

© Copyright International Business Machines Corporation 2011. All rights reserved.