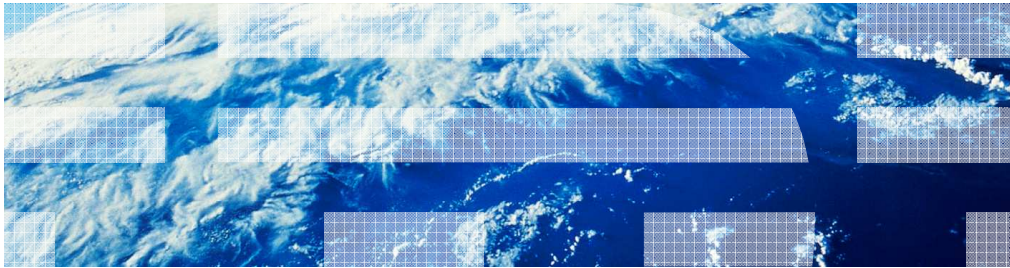




---

# IBM WebSphere Application Server V8.5

## Application Manager



WebSphere software

© 2012 IBM Corporation

This presentation describes support for the application manager included in IBM WebSphere Application Server V8.5 Liberty Profile.

# ***Overview***

This section contains an overview of the Application Manager.

## What is the application manager?

- The application manager is the feature which allows you to add, update and remove applications from the WebSphere Application Server 8.5 Liberty profile
  
- The application manager can be looked at under 3 scenarios:
  - Deploying an application
  - Updating an application
  - Removing a deployed application
  
- The application manager does not perform the tasks of deploying the applications themselves – that work is done by the application handlers
  
- All messages from the application manager will have the message key prefix “CWWKZ”

The application manager is the component in WebSphere Application Server Liberty profile responsible for initiating application installs, updates and uninstalls from the server. It does not process the install of the applications themselves – that is the role of the application handlers; the role of the application manager is purely as a portal which is used to initiate the process.

The application manager can be looked at under 3 different scenarios – deploying an application, updating an application and removing a deployed application. Deploying and removing applications have 2 different paths depending on if you use the dropin monitor or server.xml, but the updating of applications is identical no matter how it is deployed.

The application manager does not perform the tasks of deploying the applications themselves – that work is done by the application handlers. The role of the application manager is to gather the information about an application (its location, name, type and id) and to pass that information onto the application handlers to deploy. When updating applications the application manager removes then re-deploys the application in one action to ensure the application is in the correct state. When removing applications the application manager tells the necessary application handler to remove the application and then removes the application handler.

When looking in the logs for actions performed by the application manager the message key prefix used is “CWWKZ” followed by 3 numbers and a letter – the CWWKZ is the important identifier to look for as those messages originate from the application manager.

## ***Usage scenarios***

The Application Manager is used in the following scenarios.

## Deploying an application

- There are 2 ways to deploy an application to a Liberty profile server.
- The easiest is to use the dropin folder which is monitored by the dropin monitor
  - Installing using the dropin folder is quick and easy – just copy the application into the folder and the server will try to deploy the application
  - There are a few rules around file names and locations within the dropin folder that need to be followed
- The other way to install an application on a server is to use the server.xml and add an “<application>” entry into the server.xml
  - There are 4 attributes for the tag – id, name, type and location
  - By using the server.xml you have more control over the application’s deployment than you would get from using the dropin monitor

```
<application name="myApplication" type="war"
location="myAppZip.zip" />
```

There are 2 different ways to deploy an application to the application server. The quickest and easiest way to deploy an application to the server is to use the dropins folder. This folder by default is inside the server directory on the file system. You can copy an application inside the dropins folder and the server will immediately attempt to install the application (assuming the server is running – if it is not running the applications will be installed on server startup). The dropin monitor works with applications which are inside containers such as zips, wars, ears etc as well as applications that are within a directory structure. To make sure that the dropin monitor can understand enough about the application to install it you must follow some basic rules in regards to the file name and location within the dropin monitor.

The rules for using the dropins folder are as follows:

Application names are the names of the file or directory for the application that will be used as the application name and id when deployed to the server (ignoring any file types).

Location within dropins are all the applications within the dropin directory that must either be deployed in the root of the dropin directory unless the file has no type or you want to force the application to be deployed as a type different to the file type it has. If you want to force an application to be installed as a specific type you can put a directory underneath the dropin folder whose name is the type you want to force the application to be installed as. You can then place all applications you want to be forced to be that type on the server underneath that directory. A quick example would be if you have an application which is in a .zip container and you want to deploy it as a “war” application then you could put it under “dropin/war/application.zip” and the dropin monitor would install it as a war. You can put as many applications as you like under these sub-directories.

File extensions are any applications that are installed directly under the dropin folder (excluding those in the example just given) need to have a file extension. This extension is used to define the type of the application so that the server knows what to try and deploy it as. If an application does not have an extension then it will be treated as a directory which defines a type and will not work. This applies to applications installed as a single file or as a directory – both will need a “.” extension afterwards (eg “.war”).

The other way to deploy an application is to add an application entry into the server.xml for the server. You can see an example server.xml entry at the bottom of the slide with 4 attributes:

The “name” needs to be unique and depending on the application it may also be used as the context-root of the application. For a war application which is not inside a “.ear” it will be used as the context-root, but for war applications inside .ear applications then it will not be used. Eba applications define the context roots themselves so they will not use the name as the context-root. The name is also what the server uses when referring to the application in the logs.

The “type” is how you tell the server the type of the application which you are defining. We use this instead of the file extension so, as in the example, you can deploy a .zip file as a .war application without needing to change the extension type. This needs to match an application type which is supported by the server (so giving it “zip” as a type will not work, as that is not an application type).

The “location” is what the server uses to find the application you are trying to install. This can be an absolute file path - such as “C:/myApplications/anApplication.war”, a URL which we download the application from or it can also be just the filename of your application (including the file extension, if it has one). If you provide just the filename in the location field then we look for it in 2 locations - first we look inside the “apps” directory under your server’s directory (which would be “<libertyInstallLocation>/usr/servers/<yourServerName>/apps”) - if we do not find an exact filename match in that location we look in “<libertyInstallLocation>/usr/shared/apps”. If we can’t find the filename in that location as well then the server will not install the application. Note that if you deploy an application from a URL the server will not detect updates made to the application at the URL.

There is also an optional “autoStart” element you can add to the application tag – this is not required and is purely optional. If you set autoStart to “false” then the application will not start on the server until an mbean on the server is told to start the application.

Deploying an application via the server.xml allows you more customization – you can define different class loading changes to an application in the server.xml which is not possible when deploying an application in the dropin directory.

## The state machine

- The state machine is where all information about each application deployed is kept.
- The state machine ensures that each application's name is unique
- The state machine also starts an application update monitor on each application that is deployed on the server
- Any changes required to an application's state (updates, additions and removal of applications) are passed to the state machine to be processed

Once the application has been entered into the server.xml or the dropins folder then the information about the application is sent to the state machine. No matter how the application was deployed (dropin directory or server.xml entry) the application is treated almost identically from here on (any differences in the way the application is treated will be highlighted in this presentation).

As soon as an application is sent to the state machine to be deployed the state machine checks to make sure that the name of the application is not already in use – application names must be unique across the server. If it detects that another application is already deployed with the same name then it will abort the deployment and put an error message into the server logs explaining why.

Once the application manager has ensured that the application name is unique it then starts monitoring the application files for changes. This monitoring looks at all of the non-static files in the application to see if any of them are modified or deleted as well as looking for any new file additions. It ignores static files as the application does not need to be updated for the application to use the new files. From this point forward any file changes to non-static files will tell the application manager to update the application. If an application is located at a URL then the state machine will detect this and will download the application to a location within the server's cache directory before setting up the file monitoring. It will still setup a file monitor on the cached location in case of any file changes.

Once the application manager has started monitoring the application for file changes it then looks at the application type – it uses this type to find an application handler that is capable of deploying applications of that type. Once it has found an application handler which can handle the application it sends the application handler the information about the application so it can be deployed. The application manager will monitor the state of the application handler so that it knows when the application has finished deploying. Once the application is deployed then the application manager will allow updates to the application to occur (updates to the application will queue up and wait until it has finished deploying before being allowed to continue).

Any requests to add, update or remove applications are passed to the state machine from the different parts of the application manager – this is the central point for application management and controls what happens to application states and when.

## Application updates

- Application updates are looked for by an update monitor
- There is 1 update monitor per application
- The update monitor monitors all files and directories within an application for any changes (addition of files or directories as well as deletion and changes)
- Once a file change is detected the update monitor will wait until the files have stopped changing (so that file operations that take a while to complete are able to do so)
- If the update monitor detects file changes have completed and has confirmed they are to non-static files then it tells the state machine that the application has been updated

This slide covers the default behavior of the application update monitor. During application deployment the state machine will setup an application update monitor on an application. The relationship between an application update monitor and an application is 1:1 where each application will have its own update monitor.

The job of the update monitor is to monitor all the directories and files of an application for changes. It checks the files and folders every 500 milliseconds by default. If it detects that a file or directory within the application has been added, modified or removed it then checks to make sure that it is not a static file (static files are not “executed” as code and are only pointed to by an application, so changes to a static file will be reflected within the application instantly without the need to restart it). Once the update monitor has confirmed that a non-static file change has occurred it checks again to make sure that more file changes are not occurring (if copying a new file into an application it may take a while for the copy to complete depending on the system speed, so the monitor waits until no file changes occur between 2 “sweeps” of the files. Once the update monitor is happy that no more file changes are taking place it tells the state machine that the application has been updated. At this point the job of the update monitor is complete, as application updates are handled by the state machine.

## Removing a deployed application

- To remove an application deployed in the dropin folder simply delete or move it from the folder
- To remove an application installed from the server.xml simply delete or comment out the entry in the server.xml. Once the application is stopped you can then remove the application from wherever it was located on your file system.
- Once an application is removed then the update monitor will be removed for that application and any references to it in the server are cleaned up
- You can also delete an application from wherever it was installed when it has been deployed by server.xml and **not** remove the <application/> entry from the server.xml. This is not recommended but is still possible and leaves the application in a “stopped” state
- An application in the stopped state is not running on the server and cannot be accessed, but the application update monitor will continue to monitor to see if the application is re-added and the application name is still reserved in the state machine

There are several ways to remove an application from a server – these depend on how the application was deployed.

For applications deployed in the dropin folder simply remove the application from the folder – the dropin monitor will detect that the application has been removed and will tell the state machine to stop the application.

For applications deployed via the server.xml simply comment out or delete the <application> entry in the server.xml. The state machine will then remove the application from the server. Once you have seen that the application has been successfully removed in the server logs you can then delete the file from wherever it was deployed (if you want to). Once an application is removed then all references to it in the server are cleaned up completely and the update monitor is also stopped for that application.

You can also delete the application while it is still defined in the server.xml – this is not recommended but it is possible. This does not fully remove the application from the server but puts it into a “stopped” state. When an application is in a stopped state it cannot be accessed and is not running on the server, however its name is still reserved in the state machine and the application update monitor is still monitoring it’s location to see if the application is re-deployed there. To start an application which is in a stopped state simply copy it to where it is defined in the server.xml (or change the location entry for the application in the xml) and it will be started again as normal.



## Configuring the application manager

- The application manager is highly customisable
- The configuration options available are:

```
<applicationMonitor  
  pollingRate="500ms"  
  dropins="newDropinsDir"  
  dropinsEnabled="true"  
  updateTrigger="mbean"/>
```

The application manager can be highly customized. All changes to the application manager occur under the "applicationMonitor" element in the server.xml.

You can move, rename or disable the dropins folder as well as change the frequency which it checks for application addition or removal.

The "pollingRate" element sets the frequency between the dropin monitor looking for new applications or application removals, as well as how frequently the update monitor looks for changes to an application before updating it. The default value is "500ms" which is every half second.

To move or rename the dropins folder use the dropins="" element where the content of the value for "dropins" is the location and name of the new dropins folder relative to the server's directory.

The "dropinsEnabled" element is a Boolean to enable or disable adding and removing applications installed in the dropins directory. If you disable the dropins folder any applications currently deployed from the dropins folder will be removed from the server immediately.

The "updateTrigger" element allows the user to set how the server will detect updates to applications. There are 3 options:

- "polled" is the default and will scan the applications by a frequency set by the pollingRate for any changes – once a change is detected the application will be updated.
- "mbean" disables the application scanning and will activate an mbean which will only update the applications when told to by an external program to the server – the Liberty tools use mbeans to tell the server when to update applications by default.
- "disabled" completely disables application updates – the server will not update applications at all. Note that static content does not require an application to be updated by the server so any static file changes will still occur on the applications if updates are disabled.

You can have any combination of the above configuration options – if you do not set a new value for any of the above values the defaults will be used even if you override any of the other attributes.

## ***Summary***

This section contains a summary.

## Summary

- The application manager is in charge of deploying, updating and removing applications from the server
- There are 2 different ways to deploy applications – either via the dropin directory or via the server.xml
- The dropins is very quick and easy to use for simple application deployments
- The server.xml entry for an application allows more customization and lets you add in extra configuration for the application (such as custom class loading)
- Application updates are handled the same no matter how an application is deployed
- The correct way to remove an application differs depending on how it was deployed (dropins or server.xml)
- You can configure the application manager's dropin monitor and application update monitor to suit your needs

The application manager is in charge of deploying, updating and removing applications from the server. There are 2 different ways to deploy applications – either the dropin directory or the server.xml. The dropins is very quick and easy to use for simple application deployments. The server.xml entry for an application allows more customization and lets you add in extra configuration for the application (such as custom class loading). Application updates are handled the same no matter how an application is deployed. The correct way to remove an application differs depending on how it was deployed (dropins or server.xml). You can configure the application manager's dropin monitor and application update monitor to suit your needs.



## 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 V85 LP Application Manager IEA.ppt](mailto:iea@us.ibm.com?subject=Feedback%20about%20V85%20LP%20Application%20Manager%20IEA.ppt)

This module is also available in PDF format at: [../V85 LP Application Manager IEA.pdf](..V85 LP Application Manager IEA.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, and WebSphere 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>

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 2012. All rights reserved.