

# Outlining the deployment strategy

## Table of contents

<b>1</b>	<b>Introducing deployment managers</b>	<b>1</b>
<b>2</b>	<b>Resources</b>	<b>2</b>

## 1 Introducing deployment managers

Deployment managers support the automated application rollout of the application binaries and of the application configurations across platforms (that is, managing distributed and mainframe systems). Typically, a deployment manager would perform the following tasks when deploying a package into an environment:

- Verify that the package is applicable to the environment
- Sort and order the artifacts to process
- For each artifact of a given type, run the appropriate process
- Perform post deployment activities

Approval processes may be embedded. This approach will likely be necessary due to audit and legal requirements.

Deployment managers are aware of the execution environments. They generally also provide an inventory of the packages and of the current deployed binaries in a given environment. The deployment process can be implemented with scripts or with other more sophisticated techniques. These more sophisticated processes manage return codes and facilitate the use of the APIs of the different middleware systems. However, in both cases, and in order to install the package, the deployment process will consume the manifest file, which contains metadata about contents of the package.

## 2 Resources

This page contains reformatted excerpts from [Packaging and Deployment Strategies in an Open and Modern CI/CD Pipeline](#) focusing on Mainframe Software Development.