

# JBossESBDeploymentStrategies

---

## JBossESB - JBossESBDocumentation - JBossESB-4.3 Deployment Strategies

---

-

### Introduction

---

-

The JBossESB-4.3 code is packaged up and deployed as a Service ARchive (SAR) called `jbossesb.sar`. This service archive contains the core code needed for the message transport protocols and the various listener capabilities.

To make the ESB 'do' something, you can deploy services onto the ESB. The ESB is packaged with base services that ship with the ESB. A service consist of action code + configuration. Both of which should be deployed in an `.esb` archive. You can deploy as many `.esb` archives as you please. You can specify the deployment order of the archives using the `deployment.xml`. Typically you would deploy the `.esb` archive to the 'deploy' directory.

### The `.esb` archive

---

-

The idea behind the .esb archive is that it is a deployable service unit. It should enable you to move a service between servers simply by moving the .esb archive.

The standard layout of an esb archive looks like

META-INF	
	MANIFEST.MF
	jboss-esb.xml
	deployment.xml
classes	
jars	
queue-service.xml	

where,

- the jboss-esb.xml contains the service configuration (listener and actions), as well as provider configuration.  
the deployment.xml is optional, but can be used for 2 reasons:
  - make this .esb archive depend on other archives/MBeans, to specify classloading order (this includes message queues).
  - make the deployment of this .esb archive scoped.
- you custom action classes.
- additional jar archives your actions depend on.
- queue-service.xml, if your 'provider' section references queues or topics you can deploy their configuration in the .esb archive. Note that any other way to deploy these queues is fine too, we just recommend this to keep your deployments as self-contained as possible and therefore to keep dependency management simple.

JBossESB ships with a number of standard service archives, some of which are:

- [jbossesb.esb](#) - internal services like the [DeadLetterService](#),

- [jbrules.esb](#) - needed for rules evaluation using JBossRules, for services like the [Content-Based Router](#),
- [jbpm.esb](#) - needed for default provider of Business Process Management [jBPM](#)
- [smooks.esb](#) - default message transformation engine [Smooks](#).

These services are deployed by default, but you should be able to remove them if you don't need these service deployments.

## Dependencies

-

---

A number of dependencies are automatically appended to any dependencies specified in your deployment.xml configuration file. Each .esb archive automatically depends on jbossesb.esb with others being added based on the actions specified in the jboss-esb.xml configuration file.

For the current mapping see the following [list](#)

## Scoping deployment

-

---

It is possible to scope the deployment of an esb artifact, if necessary, by including a loader-repository element within the deployment.xml. More information about this element can be found in [ClassLoaderConfiguration](#), with an example of our integration available from within the [ESB qa tests](#)

## Hot Redeployment

-

---

The jbossesb-server will hot redeploy an .esb archive when the timestamp on the jar file changes, or -if the archive is exploded- when the timestamp on the META-INF/jboss-esb.xml changes .

## Examples

-

---

### 1. jbossesb.esb

The jbossesb.esb service archive contains internal services. For now it actually deploys just one service, like the [DeadLetterService](#). Most service archives depend on this service to be there. Since there is not code dependency "being there" really only means that it needs to be deployed at least once in your distributed environment.

### 2. jbrules.esb

By default the [Content-based Router](#) uses JBoss Rules to evaluate routing rules. So you need to deploy JBoss Rules to the ESB.

- The recommended way to deploy it is to deploy the jbrules.esb archive, which actually is deployed by default (so just check for it in the deploy directory). Next you can deploy your own 'myproject.esb' archive, using the actions included in the jbrules.esb.

See also the [simple\\_cbr](#) and the [fun\\_cbr](#).

### 3. jbpm.esb

If you want Business Process Management support, either for service orchestration or for human workflow support.

- The recommended deployment is to have the jbpm.esb archive deployed

See also the bpm\_orchestration quickstarts found in the [quickstarts](#) directory.