

# AvalonFAQ Merlin

## General Questions

What is Merlin?

Why was Merlin created in the first place?

Where is the official documentation?

<http://avalon.apache.org/merlin>

Where are the [JavaDocs](#)?

Typically, you don't need the [JavaDocs](#) for Merlin. You should only interact with Merlin directly if you are writing Extensions, Facilities and similar system extensions. So if you are looking for the standard interface to Merlin, it is the Avalon Framework, which can be found at <http://avalon.apache.org/framework/api/index.html>. If you **are** doing Merlin extensions, please note that the existing API has not stabilized yet, and you will have to be prepared to amend your code later for it to work. The Merlin [JavaDocs](#) are at <http://avalon.apache.org/merlin/api/index.html>.

## Building Merlin

How do I build Merlin from CVS source?

- You check out the **avalon** module from the Apache CVS repository.
- Go to the directory that was just checked out, i.e. **avalon**, and type ant

The ant script will start Maven for the five components that makes up Avalon and Merlin, namely framework, repository, meta, util and merlin. After the completion, you will find the build in **avalon/merlin/target/merlin**.

I am getting an error *unable to obtain goal [prepare-filesystem]*. What is wrong?

You don't have the Meta Plugin installed or compiled.

I get an error saying "No Goal \[avalon:meta  
]"

The two plugins need to be added to you [MAVEN-HOME]/plugins directory. The plugins are shipped with the Merlin binary distribution and include: avalon-meta-plugin-[VERSION].jar merlin-plugin-[VERSION].jar

## Running Merlin

Why does Merlin interrupt the deployment of my component?

Merlin (builds later than Jan 5, 2004) has a mechanism by which if the component takes too long to deploy, it will interrupt its deployment and move on. If your component takes a long time to initialize and deploy, use the deployment-timeout parameter in kernel.xml. Refer to: <http://avalon.apache.org/merlin/meta/kernel/parameters/index.html>

How do I define the deployment timeout a.k.a. I get a [FatalCommissioningException](#)?

In case the timeout is reached, we get a `FatalCommissioningException`:

```
org.apache.avalon.composition.model.FatalCommissioningException: target:
[[/application/main-entry]] did not respond within the timeout period:
[1000] and failed to respond to an interrupt.
```

or the like. `application/main-entry` and `1000` may vary from configuration to configuration.

The timeout may be set in a `merlin.properties` file.

```
merlin.deployment.timeout = 0
```

A value of 0 removes timeout. All other values set the timeout to milliseconds.

See <http://avalon.apache.org/merlin/merlin/kernel/properties.html> for a description where this file may be located.

Daniel Frey, 20040325

## How do I start Merlin?

## Where is the entry point?

For beginners, when you run the Merlin CLI, then that is the entry/starting point. CLI is nothing but a program with a main method that runs when you type "merlin -execute". If you want to embed Merlin into your application then you would either use CLI, [MerlinServlet](#), [MerlinBean](#) or [AbstractMerlinTestCase](#). These classes have the code that starts up merlin kernel and loads your components. If you want to get an idea what goes on behind the scenes when u run CLI then check out "IntroductionToEmbedding". The mailing list archive is also a great place ( probably the best right now, after you have read the website ) to find answers.

Vikas Phonsa

## How can there be more than one activation="startup"? It doesn't seems to matter if I take it away.

## How to I configure an NT Service? What block is launched?

Merlin uses the Java Service Wrapper to run as an NT Service. The wrapper is configured with an `wrapper.conf` file located in `%MERLIN_HOME%\bin\nt\`. The service is configured to look for an environmental variable called **MERLIN\_DAEMON** and uses this as the working directory for merlin (`merlin.dir`). You can then place a properties file in `MERLIN_DAEMON` called **merlin.properties**. A list of valid properties can be found at <http://avalon.apache.org/merlin/merlin/kernel/properties.html>. The one to notice is **merlin.deployment**. This sets the list of blocks you wish to start in your NT service.

Example file: (`%MERLIN_DAEMON%\merlin.properties`)

```
merlin.info=true
merlin.audit=true
merlin.debug=false
merlin.kernel=file:///${merlin.home}/kernel.xml
merlin.override=file:///${merlin.home}/config.xml
merlin.deployment=file:///${merlin.home}/james.xml
merlin.server=true
```

## How can I use Merlin with Eclipse?

A plugin named Merlin Developer is under development right now. Please see <http://www.merlin.softwarefabrik.biz/docs/eclipse/html/> .

But for now if you want to run the examples that come with Merlin, in Eclipse you can configure the Merlin CLI as an external tool in your IDE. I had the following configuration for it:

Tool Location: `$MERLIN_HOME\bin\merlin.bat`

Tool Argument: `classes -execute -info`

Working Directory: `${project_loc}`

I'm not following the Mavenized directory within my IDE and the "classes" in the Tool Argument refers to the directory where my class files are located, you can specify whichever directory holds your class files ( like `target/classes` if you are using mavenized directory structure ) . And you would need to place a `BLOCK-INF` folder under your source folder ("src" folder in my case) and this folder would hold your `block.xml`. Also you would need to place your `component.xinfo` files in the same folder as your component source file e.g if your component is under `tutorial` folder ( specifying "tutorial" package) the place the `xinfo` files in it. If you try to copy `BLOCK-INF` folder or the `*.xinfo` files directly to the `classes` folder, Eclipse would delete those when it rebuilds the project. But placing them under source folder would ensure the Eclipse copies that to your `classes` folder itself and also retains those there. My directory structre for a project named [MerlinTest](#) looks likes this:

```
MerlinTest/  
  src/  
    tutorial/  
      HelloComponent.java  
      HelloComponent.xinfo  
    BLOCK-INF/block.info  
  classes/  
    tutorial/  
      HelloComponent.class  
      HelloComponent.xinfo  
    BLOCK-INF/block.info
```

Now you should be able to run Merlin CLI from within eclipse by selecting your project (like [MerlinTest](#) in my case) in the left navigation or project explorer and go to Run->External Tools and just run the Merlin CLI.

And then you can run and test your components in the same way.

It was a little painful for me to make this work initially so I hope this helps a little.

Vikas Phonsa.

## Extensions

### How do I embedd Merlin into another application?

See the [IntroductionToEmbedding](#) page for more and detailed information.

### What kind of extensions are available?

### What is the difference between an Extension and a Facility?

### How do I run the [AbstractMerlinTestCase](#) from Ant?

See the [AbstractMerlinTestCase](#) page for more and detailed information.

### Where I can find description of classloader hierarchy in merlin?

