

# About Apache Excalibur

Apache Excalibur spun out of [Apache Avalon](#) in 2004. Excalibur continues to develop the basic Avalon Framework and Excalibur platform including the Fortress IoC container.

## Excalibur Explained

Excalibur is a platform for component and container applications built on key design patterns such as Inversion of Control and Separation of Concerns available for the Java platform. You can use Excalibur to build any number of other applications from desktop centric Swing apps to complicated servers. The Avalon Framework (the core of Excalibur) is often used as the "underpinning" of a larger application or platform. Such applications include Apache Cocoon, Apache James, and the Keel Framework.

## What Excalibur Isn't

Excalibur is not part of the J2EE stack. While Excalibur can be used in developing J2EE applications and even used to build a J2EE server, the platform is not part of J2EE or a J2EE application server. Excalibur is much simpler and does not include many J2EE specifications out of the box. That said, you might write a J2EE application which embeds Excalibur (such as in a servlet) or you may embed a J2EE server in a larger Excalibur-based container.

## What Excalibur Is

The Excalibur platform is the following:

- A core framework for component programming (The Avalon Framework)
- An IoC container called Fortress
- A set of container utilities called [ContainerKit](#)
- A component library called Cornerstone

When working with Excalibur you may use one or all of these. For more information on makeup of Excalibur, see the [Excalibur Overview](#).

## Who Uses Excalibur?

Excalibur users generally find themselves in one or more of the following roles:

- **Excalibur User:** You are using another application which uses Excalibur or Avalon "under the hood." Perhaps you are extending this application or are simply a user who wants to better understand the inner workings.
- **Component Author:** You are developing a component to be run in an Excalibur or Avalon container. You may be assembling a set of components into a larger application block and then deploying that block and the container (Fortress) as a final complete application.
- **Container Developer:** You are extending an existing Excalibur or Avalon container or developing a new Avalon compliant container.

Regardless of your role, there are certain technologies and concepts which are essentially prerequisites to Excalibur development and use. If you are not familiar with these terms, please read the corresponding documentation:

- [Java Platform](#)
- [InversionOfControl](#)
- [SeparationOfConcerns](#)
- [ComponentOrientedProgramming](#)

## Why Use Excalibur?

First we need to answer a more basic question: why would I use a container? How does a container fit into my application?

...  
[fixme]

To understand why someone would use Excalibur, let's look at some of the problems Excalibur solves.

- Component Coupling
- Dependency Management
- Implementation Lock-in
  
- Lifecycle Management  
... [fixme]

## How to Use Excalibur?

This is a question for another article: check out our [FiveMinuteTutorial](#).

## References

For more information, we recommend these articles:

- [ExcaliburOverview](#)
- [WhichContainer](#)
- [ExcaliburHistory](#)