

---

**Overview Package Class Tree Index Help**[PREV CLASS](#) [NEXT CLASS](#)[SUMMARY: NESTED | FIELD | CONSTR | METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)[DETAIL: FIELD | CONSTR | METHOD](#)

---

**edu.mit.csail.aeolus.api**  
**Interface AeolusSafe****All Known Implementing Classes:**

[AeolusBox](#), [AeolusClosure](#), [AeolusLock](#), [AeolusQueue](#), [AeolusSequence](#), [AeolusShared](#),  
[AeolusTag](#), [EventID](#), [PID](#)

---

```
public interface AeolusSafe
```

AeolusSafe is a marker interface for classes that are safe to share between threads without copying. In order to be AeolusSafe, a class must either extend AeolusShared, or it must contain no mutable state. All subclasses of the class must also be AeolusSafe. An AeolusSafe class that is not AeolusShared is must meet the following requirements, which are verified at load time.

- All of its fields must be declared `final`.
  - The types of its fields must also implement AeolusSafe, or be primitive types (other than arrays), i.e., they all must be "Safe to Share."
  - No inner classes are allowed.
- 
- 

**Overview Package Class Tree Index Help**[PREV CLASS](#) [NEXT CLASS](#)[SUMMARY: NESTED | FIELD | CONSTR | METHOD](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)[DETAIL: FIELD | CONSTR | METHOD](#)

---