



Jini™ Entry Utilities Specification

A Jini™ technology entry provides a way to store a collection of related objects in a way amenable to simple exact-match searches. When designing entries, certain tasks are commonly done in similar ways. This specification defines a utility class for such common tasks.



Version 1.1Alpha
November 1999

Copyright © 1999 Sun Microsystems, Inc.
901 San Antonio Road, Palo Alto, CA 94303 USA.
All rights reserved.

Sun Microsystems, Inc. has intellectual property rights (“Sun IPR”) relating to implementations of the technology described in this publication (“the Technology”). In particular, and without limitation, Sun IPR may include one or more patents or patent applications in the U.S. or other countries. Your limited right to use this publication does not grant you any right or license to Sun IPR nor any right or license to implement the Technology. Sun may, in its sole discretion, make available a limited license to Sun IPR and/or to the Technology under a separate license agreement. Please visit <http://www.sun.com/software/communitysource/>.

Sun, the Sun logo, Sun Microsystems, Jini, the Jini logo, JavaSpaces, Java, JavaBeans, Solaris, NFS, PC-NFS, EmbeddedJava, PersonalJava, and Solstice are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

THIS SPECIFICATION IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS SPECIFICATION COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THE SPECIFICATION. SUN MICROSYSTEMS, INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN ANY TECHNOLOGY, PRODUCT, OR PROGRAM DESCRIBED IN THIS SPECIFICATION AT ANY TIME.

Contents

1	Entry Utilities	1
1.1	AbstractEntry	1
1.2	Serialized Form	2
1.3	Comments	2

The Jini™ Entry Utilities Specification

EU.1 Entry Utilities

ENTRIES are designed to be used in distributed algorithms for which exact-match lookup semantics are useful. An entry is a typed set of objects, each of which may be tested for exact match with a template. The details of entries and their semantics are discussed in the *Jini Entry Specification*.

When designing entries, certain tasks are commonly done in similar ways. This specification defines a utility class for such common tasks.

EU.1.1 AbstractEntry

The class `net.jini.entry.AbstractEntry` is a specific implementation of `Entry` that provides useful implementations of `equals`, `hashCode`, and `toString`:

```
package net.jini.entry;

public abstract class AbstractEntry implements Entry {
    public boolean equals(Object o) {...}
    public int hashCode() {...}
    public String toString() {...}
    public static boolean equals(Entry e1, Entry e2) {...}
    public static int hashCode(Entry entry) {...}
    public static String toString(Entry entry) {...}
}
```

The static method `AbstractEntry.equals` returns `true` if and only if the two entries are of the same class and for each field F , the two objects' values for F are either both `null` or the invocation of `equals` on one object's value for F with the other object's value for F as its parameter returns `true`. The static method `hashCode` returns zero XOR the `hashCode` invoked on each non-`null` field of the entry. The static method `toString` returns a string that contains each field's name and value. The non-static methods `equals`, `hashCode`, and `toString` return a result equivalent to invoking the corresponding static method with `this` as the first argument.

EU.1.2 Serialized FormComments

Class	serialVersionUID	Serialized Fields
AbstractEntry	5071868345060424804L	<i>none</i>

EU.1.3 Please direct comments to jini-comments@java.sun.com.