

JiniTM Entry Utilities Specification

A JiniTM 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.



THE NETWORK IS THE COMPUTER[®]

901 San Antonio Road
Palo Alto, CA 94303 USA
415 960-1300
fax 415 969-9131

Revision 1.0
January 25, 1999

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

All rights reserved. Copyright in this document is owned by Sun Microsystems, Inc.

Sun Microsystems, Inc. has patent and other intellectual property rights relating to implementations of the technology described in this Specification ("Sun IPR"). Your limited right to use this Specification does not grant you any right or license to Sun IPR. A limited license to Sun IPR is available from Sun under a separate Community Source License.

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. SUN SHALL NOT BE LIABLE FOR ANY DAMAGES SUFFERED BY YOU AS A RESULT OF USING THE SPECIFICATION.

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 THE SPECIFICATIONS AT ANY TIME, IN ITS SOLE DISCRETION. SUN IS UNDER NO OBLIGATION TO PRODUCE FURTHER VERSIONS OF THE SPECIFICATION OR ANY PRODUCT OR TECHNOLOGY BASED UPON THE SPECIFICATION. NOR IS SUN UNDER ANY OBLIGATION TO LICENSE THE SPECIFICATION OR ANY ASSOCIATED TECHNOLOGY, NOW OR IN THE FUTURE, FOR PRODUCTIVE OR OTHER USE.

RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure by the U.S. Government is subject to restrictions of FAR 52.227-14(g)(2)(6/87) and FAR 52.227-19(6/87), or DFAR 252.227-7015(b)(6/95) and DFAR 227.7202-1(a).

TRADEMARKS

Sun, the Sun logo, Sun Microsystems, Jini, JavaSpaces, JavaSoft, JavaBeans, JDK, Java, HotJava, HotJava Views, Visual Java, Solaris, NEO, Joe, Netra, NFS, ONC, ONC+, OpenWindows, PC-NFS, EmbeddedJava, PersonalJava, SNM, SunNet Manager, Solaris sunburst design, Solstice, SunCore, SolarNet, SunWeb, Sun Workstation, The Network Is The Computer, ToolTalk, Ultra, Ultracomputing, Ultraserer, Where The Network Is Going, Sun WorkShop, XView, Java WorkShop, the Java Coffee Cup logo, and Visual Java are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Contents



| | |
|---------------------------------|----------|
| 1. Entry Utilities | 1 |
| 1.1 AbstractEntry | 1 |
| 1.2 Serialized Form | 2 |



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.

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.

1.2 *Serialized Form*

The `serialVersionUID` of `AbstractEntry` is 5071868345060424804L. There are no serialized fields.