Status for Java Developers Kit API for IPv6

Status of This Memo

This memo provides information to the Grid community regarding the current ability of the Java
Developers Kit API to support IPv6 functionality as provided in the IETF RFC 3493 Basic Socket
Extensions for IPv6 API.   It does not define any standards or technical recommendations.
Distribution of this memo is unlimited.

Abstract

This document serves two functions. State the ability of the Java Developers Kit API for IPv6 to
support the functionality within IETF RFC 3493 Basic Socket Extensions for IPv6. Then features
that are required within the Java Developers Kit API to support additional IPv6 features in RFC
3493.

Contents

1. Introduction

The goal of this document is to help the reader understand the capabilities of IPv6 using the Java Developers Kit (JDK). IETF RFC 3493 is the widely implemented Basic API for IPv6, but assumes a BSD UNIX sockets model as the interface to IPv6. The programming model for sockets is different than the programming model for the Java Development Kit (JDK) to communicate with the Internet Protocol suite on a platform, but not in basic TCP/IP functions supported. RFC 3493 is a C Programming centric model and a one to one mapping of RFC 3493 to JDK is not a practical view. This document will state the capabilities of the JDK to support IPv6 as of JDK 1.4, and then IPv6 features required for future JDK versions are described.

2. Java API IPv6 Capabilities

JDK 1.4 java.net package supports a range of methods to program networking functions into an application within the java.net package factory interfaces and classes for programming TCP/IP applications. JDK 1.4 supports the same capabilities as defined in RFC 3493. This is supported by use of the class InetAddress, which supports Inet4Address and Inet6Address classes. This permits the use of methods to access data based on those addresses with a series of get*() methods, and Boolean operators which permit the same capabilities as defined in RFC 3493 within the Java programming model.

RFC 3493 permits a request through the getaddrinfo () API the option to inform the name to address module to return IPv4 addresses as IPv6 IPv4-Mapped Addresses, which is simply an IPv6 representation of an IPv4 address within a 128bit IPv6 address format. This permits applications to treat all addresses as IPv6 and redirect them to the network interface as IPv4 or IPv6 addresses within the TCP/IP implementation.

There is no similar API syntax within java.net InetAddress class to return IPv4-Mapped addresses or within the methods to use the Inet4Address class. But, Java does support IPv4-Mapped addresses semantically within java.net and can return IPv4-Mapped addresses internally to a TCP/IP implementation platform as a feature within the Java programming environment. Most applications will simply use the InetAddress class, and it will be transparent to a user within the Java programming model, the support of IPv4-Mapped addresses within the TCP/IP implementation.

See JDKv6 reference below in this document. Other systems properties for JDK and IPv6 can be found at the following URLs for systems implementers using IPv6:
http://java.sun.com/j2se/1.4.2/docs/guide/net/ipv6_guide/#ipv6-related
http://java.sun.com/j2se/1.4.2/docs/api/

New features for JDK 1.5 that will benefit IPv6 will be officially documented soon.

3. Java API Additional Requirement

The only two IPv6 functions required to be supported within RFC 3493 is setting the IPv6 Flow Label and supporting the V6_ONLY option to never permit IPv4-Mapped addresses. These requirements are understood for the JDK and are being looked at now as to how they will be supported in the future.

Any new changes or requirements for the Java API at this time will have to come from JDK 1.6.

This document does not consider the Advanced Socket API (RFC 3542) for the Java API.

4.   Security Considerations

This document is informational, providing information on the use of IPv6 in Java coding of GGF specifications. It does not in itself have any security implications.

Author Information

Jim Bound
Hewlett-Packard Company
110 Spitbrook Road ZKO3-3/W20
Nashua, NH 03062,
USA
EMail: Jim.Bound@hp.com
Phone: +1-603-884-0062

Intellectual Property Statement

The GGF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the GGF Secretariat.

The GGF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this recommendation. Please address the information to the GGF Executive Director.

This document and the information contained herein is provided on an "AS IS" basis and THE GLOBAL GRID FORUM DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE."

References

[JDKv6] *Java Development Kit 1.4.1, IPv6 Guide*,
        http://java.sun.com/j2se/1.4.1/docs/guide/net/ipv6_guide

[RFC3493] R. Gilligan, S. Thompson, J. Bound, J. McCann, W. Stevens, *Basic Socket Interface Extensions for IPv6*, IETF RFC (obsoletes RFC2553), February 2003.

[RFC3542] W. Stevens, M. Thomas, E. Nordmark, T Jinmei, *Advanced Socket Application Program Interface (API) for IPv6* IETF RFC3542 (obsoletes RFC2292), May 2003