



| IBM Software Group

64-bit support

Java and JNI considerations



@business on demand.

© 2007 IBM Corporation
Updated May 14, 2015

This presentation will show Java™ and JNI changes in a 64-bit system on WebSphere Base Application Server V6.1 on z/OS®

Agenda

- JVM version and code changes
- Macros and JNI calls



This brief presentation will discuss JVM version and code changes, and macros and JNI calls.

Section

JVM version and code changes

This section will discuss JVM version and code changes.

Details

- 64bit uses J2RE 1.5.0 IBM z/OS
- Java code update (no major update except some JNI call changes)



J2RE 1.5.0 IBM z/OS is used. There are 2 JVMs, 31-bit and 64-bit for different runtime modes in WebSphere Application Server Version 6.1.

No major Java code change has been done to be 64-bit compliant.

Some code has been modified where the `loadlibrary(..)` call is needed to load `bbg`, 64-bit modules, rather than `bbo`, 31-bit modules based on JVM runtime bit mode.

There were some changes done when using Java-to-native and native-to-java calls passing native addresses, this will be discussed in the later part of this presentation.

Section

Macro and JNI calls



This section will discuss Macro and JNI calls.

Passing pointers between native and Java code

- ***inline long long setNativeAddressTo_jlong (long NativeAddress);***

Example:

```
CORBA::Request * the_request;
... ..
jlong native_the_request =
(jlong) setNativeAddressTo_jlong ((long) the_request);
jargs[3].j = (jlong) native_the_request;
```

6

Java and JNI considerations

© 2007 IBM Corporation

To keep code common, between 31-bit and 64-bit native, macros have been introduced such as jlong that will be used to pass addresses into native-to-java and java-to-native calls regardless of bit mode. As jlong is always 8 bytes, regardless of mode, the lower 4 bytes will be used to store the address under 31-bit mode and all 8 bytes will be used under 64-bit mode. The macros will expand accordingly based on compile mode.

setNativeAddressTo_jlong(..) takes an address to store in jlong before passing it in a native-to-java call.

Passing pointers (cont.)

- ***inline unsigned long getNativeAddressFrom_jlong
(long long in_jlong);***

Example:

```
extern "C" JNIEXPORT jint
Java_com_ibm_ws390_orb_ClientDelegate_jorbrelease(JNIEnv * env, jobject
object, jlong bboojorb_ptr)
{
    long native_bboojorb_ptr = (long)getNativeAddressFrom_jlong((long
long)bboojorb_ptr);
    .....
}
```



getNativeAddressTo_jlong(..) retrieves an address from jlong (passed to native) according to the bit mode.

If a structure with a native pointer is being passed between native and Java as a byte array, the proper offset should be calculated to retrieve the address based on the bit mode. This is done because alignment of the pointer data type changes in 64-bit mode.

Trademarks, copyrights, and disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM z/OS

Java, JNI, JVM, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products; their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2007. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.