(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
14 November 2002 (14.11.2002)

PCT

(10) International Publication Number
WO 02/091173 A2

(51) International Patent Classification[7]: G06F 9/44

(21) International Application Number: PCT/US02/14584

(22) International Filing Date: 8 May 2002 (08.05.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/852,402 9 May 2001 (09.05.2001) US

(71) Applicant: **SUN MICROSYSTEMS, INC.** [US/US]; M/S: UPAL01-521, 901 San Antonio Road, Palo Alto, CA 94303 (US).

(72) Inventors: **SOKOLOV, Stepan**; 34832 Dorado Common, Fremont, CA 94555 (US). **WALLMAN, David**; 777 S. Mathilda Avenue, #266, Sunnyvale, CA 94087 (US).

(74) Agent: **VILLENEUVE, Joseph, M.**; BEYER WEAVER & THOMAS, LLP, 2030 Addison Street, 7th Floor, P.O. Box 778, Berkeley, CA 94704 (US).

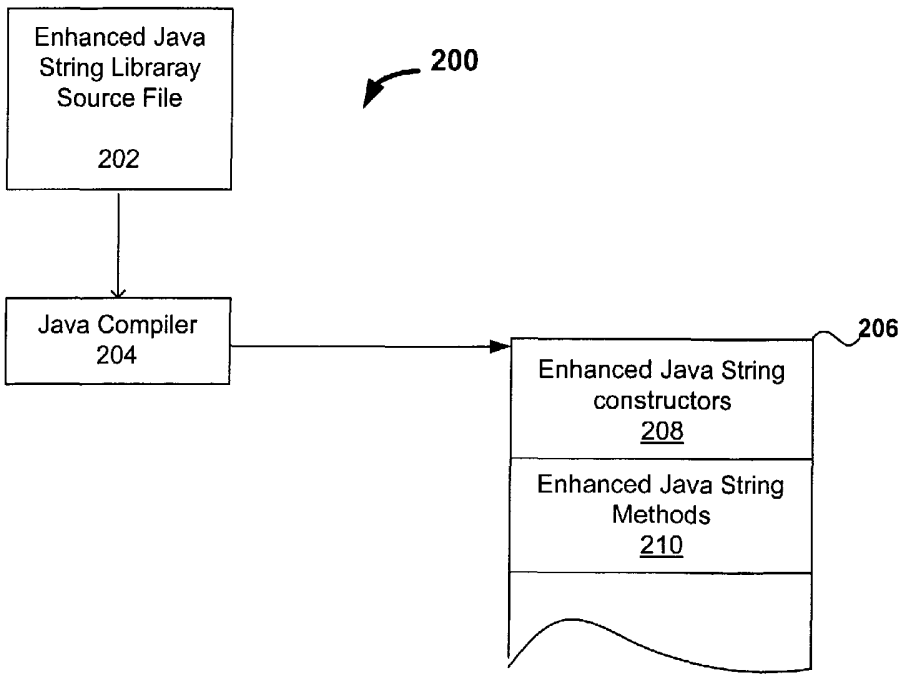
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

[Continued on next page]

(54) Title: FRAMEWORKS FOR EFFICIENT REPRESENTATION OF STRING OBJECTS IN JAVA PROGRAMMING ENVIRONMENTS



(57) Abstract: Alternative techniques for representation of Java string objects are needed. The techniques are especially useful for representing Java objects in Java computing environments and can thereby improving the performance of a virtual machine, especially those with relatively limited resources (e.g., embedded systems with relatively smaller memory and computing power).

WO 02/091173 A2

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

FRAMEWORKS FOR EFFICIENT REPRESENTATION OF STRING OBJECTS IN JAVA PROGRAMMING ENVIRONMENTS

BACKGROUND OF THE INVENTION

The present invention relates generally to object-based high level programming environments, and more particularly, to techniques for representing string objects in object-based computing environments.

Recently, the Java programming environment has become quite popular. The Java programming language is a language that is designed to be portable enough to be executed on a wide range of computers ranging from small devices (e.g., pagers, cell phones and smart cards) up to supercomputers. Computer programs written in the Java programming language (and other languages) may be compiled into Java Bytecode instructions that are suitable for execution by a Java virtual machine implementation.

The Java virtual machine is commonly implemented in software by means of an interpreter for the Java virtual machine instruction set but, in general, may be software, hardware, or both. A particular Java virtual machine implementation and corresponding support libraries together constitute a Java runtime environment.

Computer programs in the Java programming language are arranged in one or more classes or interfaces (referred to herein jointly as classes or class files). Such programs are generally platform, i.e., hardware and operating system, independent. As such, these computer programs may be executed without modification on any computer that is able to run an implementation of the Java runtime environment.

Object-oriented classes written in the Java programming language are compiled to a particular binary format called the "class file format." The class file includes various components associated with a single class. These components can be, for example, methods and/or interfaces associated with the class. In addition, the class file format can include a significant amount of ancillary information that is associated with the class.

The class file format (as well as the general operation of the Java virtual machine) is described in some detail in The Java Virtual Machine Specification, Second Edition, by Tim Lindholm and Frank Yellin, which is hereby incorporated herein by reference.

As an object-oriented programming language, Java utilizes the programming concept known as an object. In the Java programming environment, Java objects are created (instantiated) from Java classes. One common Java object is the Java string object which can represent a string of one or more characters. Conventionally, representation of each character in the string requires two bytes of data to be used. Thus, Java string objects are represented in arrays of two-byte characters (i.e., a series of entries where each entry contains a two-byte value representing a character).

As will be appreciated, the conventional representation of Java objects is inefficient since there is not always a need to represent strings in arrays of two-byte characters. In fact, the conventional approach can be grossly inefficient since, in most cases, there may not be a need to represent strings in arrays of two-byte characters. Thus, the conventional representation of Java string objects can seriously hinder the performance of a virtual machine, especially those with relatively limited resources (e.g., embedded systems with relatively smaller memory and computing power).

In view of the foregoing, alternative representation of Java string objects are useful.

SUMMARY OF THE INVENTION

As noted in the background of the invention, alternative techniques for representation of Java string objects are needed. Broadly speaking, the present invention relates to improved techniques for representing string objects in an object-oriented environment. The techniques are especially useful for representation of Java objects and can thereby improving the performance of a virtual machine, especially those with relatively limited resources (e.g., embedded systems with relatively smaller memory and computing power).

In accordance with one aspect of the invention, techniques for instantiating Java string objects are disclosed. The techniques can be implemented to create Java objects as arrays of one-byte characters when it is appropriate. These techniques can be utilized by an enhanced constructor that creates Java objects in an efficient manner. The constructor can be provided in a Java library that is available to application programs (or programmers). Another aspect of the invention provides for enhanced Java methods that can perform operations on Java objects represented as arrays of one-byte characters. The enhanced Java methods can also be provided in the Java library.

The invention can be implemented in numerous ways, including as a method, an apparatus, a computer readable medium, and a database system. Several embodiments of the invention are discussed below.

These and other aspects and advantages of the present invention will become more apparent when the detailed description below is read in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will be readily understood by the following detailed description in conjunction with the accompanying drawings, wherein like reference numerals designate like structural elements, and in which:

Fig. 1 represents a Java string object in accordance with one embodiment of the invention.

Fig. 2 represents a Java computing environment in accordance with one embodiment of the invention.

Fig. 3 illustrates a method for instantiating a Java string object in accordance with one embodiment of the invention.

Fig. 4 illustrates a method of performing one or more operations on a Java string object in accordance with one embodiment of the invention.

DETAILED DESCRIPTION OF THE INVENTION

As noted in the background of the invention, alternative techniques for representation of Java string objects are needed. The present invention pertains to improved techniques for representing string objects in an object-oriented environment. The techniques are especially useful for representing Java objects in Java computing environments and can thereby improving the performance of a virtual machine, especially those with relatively limited resources (e.g., embedded systems with relatively smaller memory and computing power).

In accordance with one aspect of the invention, techniques for instantiating Java string objects are disclosed. The techniques can be implemented to create Java objects as arrays of one-byte characters when it is appropriate. These techniques can be utilized by an enhanced constructor that creates Java objects in an efficient manner. The constructor can be provided in a Java library that is available to application programs (or programmers). Another aspect of the invention provides for enhanced Java methods that can perform operations on the Java objects represented as arrays of one-byte characters. The enhanced Java methods can also be provided in the Java library.

Embodiments of the invention are discussed below with reference to Figs. 1-4. However, those skilled in the art will readily appreciate that the detailed description given herein with respect to these figures is for explanatory purposes only as the invention extends beyond these limited embodiments.

Fig. 1 represents a Java string object 100 in accordance with one embodiment of the invention. The Java string object 100 includes a length field 102 and a reference field 104. The length field 102 represents the length of the characters in the string object. The reference field 104 represents a reference to a memory portion 106 where these characters are stored (e.g., Byte 1- Byte N). As shown in Fig. 1, in addition to the characters stored in Byte 1 through Byte N, the memory portion 106

includes a descriptor field 108. The descriptor field 108 can indicate that this array object is stored as an array of one-byte characters.

As will be appreciated, an application program can allocate Java objects as arrays of one-byte characters in accordance with one embodiment of the invention. Moreover, this can be accomplished without requiring the application program to perform any additional work. In other words, the invention provides for creation of a string object as an array of one-byte characters when appropriate and without requiring the application program (or Java programmer) to explicitly allocate the string object as an array of one-byte characters. To achieve this and other objectives of the invention, an enhanced Java string constructor is provided in accordance with one embodiment of the invention. The enhanced Java string constructor can be provided in a Java library which may also include one or more enhanced Java string methods suitable for performing various operations on Java string objects that have been allocated as arrays of one-byte characters (i.e., each Java object is represented as an array of one-byte characters.

To illustrate, Fig. 2 represents a Java computing environment 200 in accordance with one embodiment of the invention. As shown in Fig. 2, the Java computing environment 200 includes an enhanced Java string library source file 202 which is input to a Java compiler 204. The enhanced Java string library source file 202 is compiled by the Java compiler 204 and an enhanced Java string class file is produced which can be provided as a Java library to application programs (or programmers).

The Java string class file 206 can provide one or more constructors 208 suitable for instantiating Java string objects that are allocated as arrays of one-byte characters (i.e., each Java object is represented as an array of one-byte characters). In addition, one or more enhanced Java string methods 210 can be provided in the Java string class file 206. As will be appreciated, the one or more enhanced Java string methods 210 can perform one or more operations on Java string objects represented as arrays of one-byte characters. These operations can include the operations performed by conventional Java string methods operating on Java string

objects represented as arrays of two-byte characters. These conventional operations are known to those skilled in the art. In one embodiment, the one or more enhanced Java string methods can perform these operations on Java objects represented as arrays of two-byte characters as well as those objects represented as arrays of one-byte characters.

As noted above, an enhanced Java string constructor can be used to instantiate Java string objects. Fig. 3 illustrates a method 300 for instantiating a Java string object in accordance with one embodiment of the invention. As such, the method 300 can, for example, be performed by an enhanced Java string constructor. Initially, at operation 302, one or more characters associated with a Java string object are received as input. These one or more characters can, for example, be provided by an application program that seeks to instantiate a Java string as a string of one or more characters. Next, at operation 304, a determination is made as to whether an array of one-byte characters or an array of two- byte characters should be allocated to represent said Java string object. In one embodiment, a determination is made as to whether all the characters can appropriately be represented in one byte.

If it is determined at operation 304 that an array of one-byte characters should be allocated, the method 300 proceeds to operation 306 where an array of one-byte characters is allocated to represent each of the characters of the Java string as a single byte. The method 300 ends following operation 306. However, if it is determined at operation 304 that an array of two-byte characters should be allocated, the method proceeds to operation 308 where an array of two-byte characters is allocated to represent each of the characters of the Java string in two bytes. The method 300 ends following operation 308.

Fig. 4 illustrates a method of performing one or more operations on a Java string object in accordance with one embodiment of the invention. The method 400 can, for example, be performed by the one or more enhanced Java string methods 210 of Fig. 2. Initially, at operation 402, a request to perform an operation on a Java string object is received. Next, at operation 404, a determination is made as to whether the Java string

object has been allocated as an array of one-byte characters or as an array of two-byte characters. In one embodiment, this determination is made by reading a descriptor associated with the array (e.g., descriptor field 108 of Fig. 1).

If it is determined at operation 404 that the Java string object has been allocated as an array of one-byte characters, the method 400 proceeds to operation 406 where an appropriate method suitable for performing the operation on Java string objects that are allocated as arrays of one-byte characters is invoked. However, if it is determined at operation 404 that the Java string object has been allocated as an array of two-byte characters, the method 400 proceeds to operation 408 where an appropriate method suitable for performing the operation on Java string objects that are allocated as arrays of two-byte characters is invoked.

It should be noted that the method invoked in operation 408 can be a conventional method while the method invoked in operation 406 is an enhanced Java method that can operate on Java string objects represented as arrays of one-byte characters. It should also be noted that, in a preferred embodiment, the enhanced Java method can operate on Java string objects represented as arrays of one-byte characters or as arrays of two-byte characters. Thus, only one enhanced Java string method need be invoked to perform the operation on the Java string object. To achieve this, the enhanced Java string method can perform the determination made at operation 404 and then proceed accordingly based on that determination.

The many features and advantages of the present invention are apparent from the written description, and thus, it is intended by the appended claims to cover all such features and advantages of the invention. Further, since numerous modifications and changes will readily occur to those skilled in the art, it is not desired to limit the invention to the exact construction and operation as illustrated and described. Hence, all suitable modifications and equivalents may be resorted to as falling within the scope of the invention.

What is claimed is:

CLAIMS

1. In a Java programming environment, a method of instantiation of a Java string object, said method comprising:

receiving one or more characters;

determining whether an array of one-byte characters or an array of two- byte characters should be allocated to represent said Java string object;

allocating an array of one-byte characters to represent said Java string object when said determining determines that said Java string object should be allocated as an array of one-byte characters; and

allocating an array of two-byte characters to represent said Java string object when said determining determines that said Java string object should be allocated as an array of two-byte characters.

2. A method as recited in claim 1, wherein said determining is performed by a Java constructor.

3. A method as recited in claim 1, wherein allocation of said string object is performed by a Java constructor.

4. A method as recited in claim 1,

wherein said determining of whether an array of one-byte characters or an array of two- byte characters should be allocated to represent said Java string object is performed by a Java constructor; and

wherein allocation of said string object is performed by a Java constructor.

5. A method as recited in claim 4, wherein said Java constructor is provided in a Java library.

6. A method as recited in claim 5, wherein said Java library further comprises:

one or more Java methods suitable for performing operations on said Java object.

7. In a Java programming environment, a method of performing one or more operations on a Java string object, said method comprising:

receiving a request to perform an operation on a Java string object;

determining whether said Java string object has been allocated as an array of one-byte characters or as an array of two-byte characters;

invoking a method suitable for performing said operation on Java string objects that are allocated as arrays of one-byte characters when said determining determines that the said Java string object has been allocated as a array of one-byte characters.

8. A method as recited in claim 7,

wherein said method is also suitable for performing said operation on Java string objects allocated as arrays of two-byte characters, and

wherein said method is also invoked when said determining determines that the said Java string object has been allocated as an array of two-byte characters.

9. A method as recited in claim 7, wherein said method further comprises:

invoking another method suitable for performing said operation on Java string objects that are allocated as arrays of two-byte characters when said determining determines that the said Java string object has been allocated as a array of two- byte characters.

10. A method as recited in claim 7, wherein said determining of whether a Java string object has been allocated as an array of one-byte characters or as an array of two-byte characters is performed by a Java method.

11. A method as recited in claim 10, wherein said Java method is provided in a Java library.

12. A method as recited in claim 7, wherein said Java method is provided in a Java library.

13. A method as recited in claim 12, wherein said Java library further comprises:

one or more constructors associated with said Java string object, said one or more constructors being suitable for allocation of said Java string object.

14. A method as recited in claim 7, wherein said determining of whether a Java string object has been allocated as an array of one-byte characters or as an array of two-byte characters comprises:

reading a descriptor of said array, said descriptor indicating whether said Java string object is represented as an array of one-byte characters or as an array of two-byte characters.

15. A Java computing environment, wherein said environment comprises:

a Java string object suitable for representing a Java string of one or more characters, wherein said Java string is represented in an array of one byte characters in a memory portion of said Java computing environment.

16. A Java computing environment as recited in claim 15, wherein said environment further comprises:

at least one Java constructor suitable for instantiating said Java string object; and

at least one Java method suitable for performing one or more operations on said Java string object.

17. A Java computing environment as recited in claim 15, wherein said at least one Java constructor and said at least one Java method are provided in a Java library.

18. A computer readable media including computer program code for a Java library, said computer readable media comprising:

computer program code for at least one Java constructor suitable for instantiating said Java string object; and

wherein said computer program code for said at least one Java constructor can operate to allocate a Java string object as an array of one-byte characters.

19. A computer readable media as recited in claim 18, further comprising:

computer program code for at least one Java method, and

wherein said program code for said at least one Java method is capable of performing one or more operations on said Java string object represented as an array of one-byte characters.

20. A computer readable media as recited in claim 18, wherein said computer program code for said method is also capable of performing one or more operations on Java string objects represented as arrays of two-byte characters.

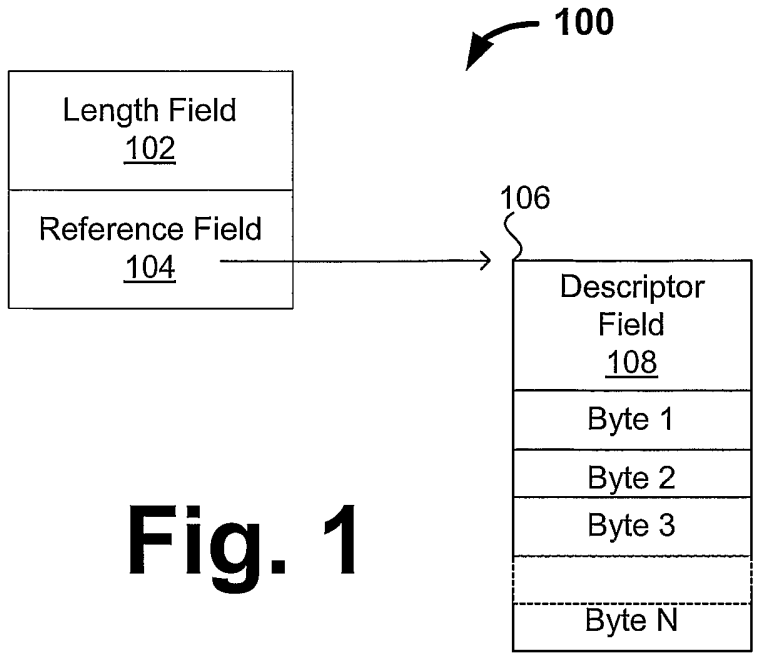

100
Length Field
102
Reference Field
104
106
Descriptor
Field
108
Byte 1
Byte 2
Byte 3
Byte N

Fig. 1

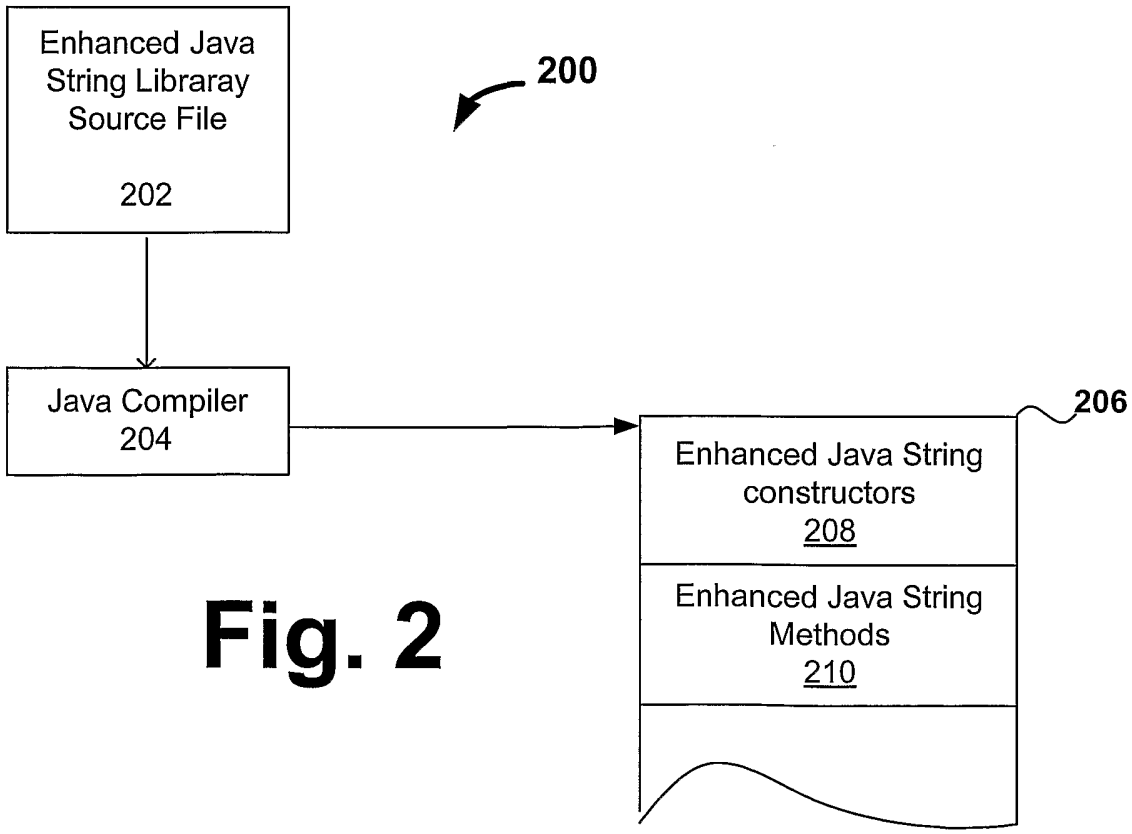

Enhanced Java
String Libraray
Source File
202
200
Java Compiler
204
206
Enhanced Java String
constructors
208
Enhanced Java String
Methods
210

Fig. 2

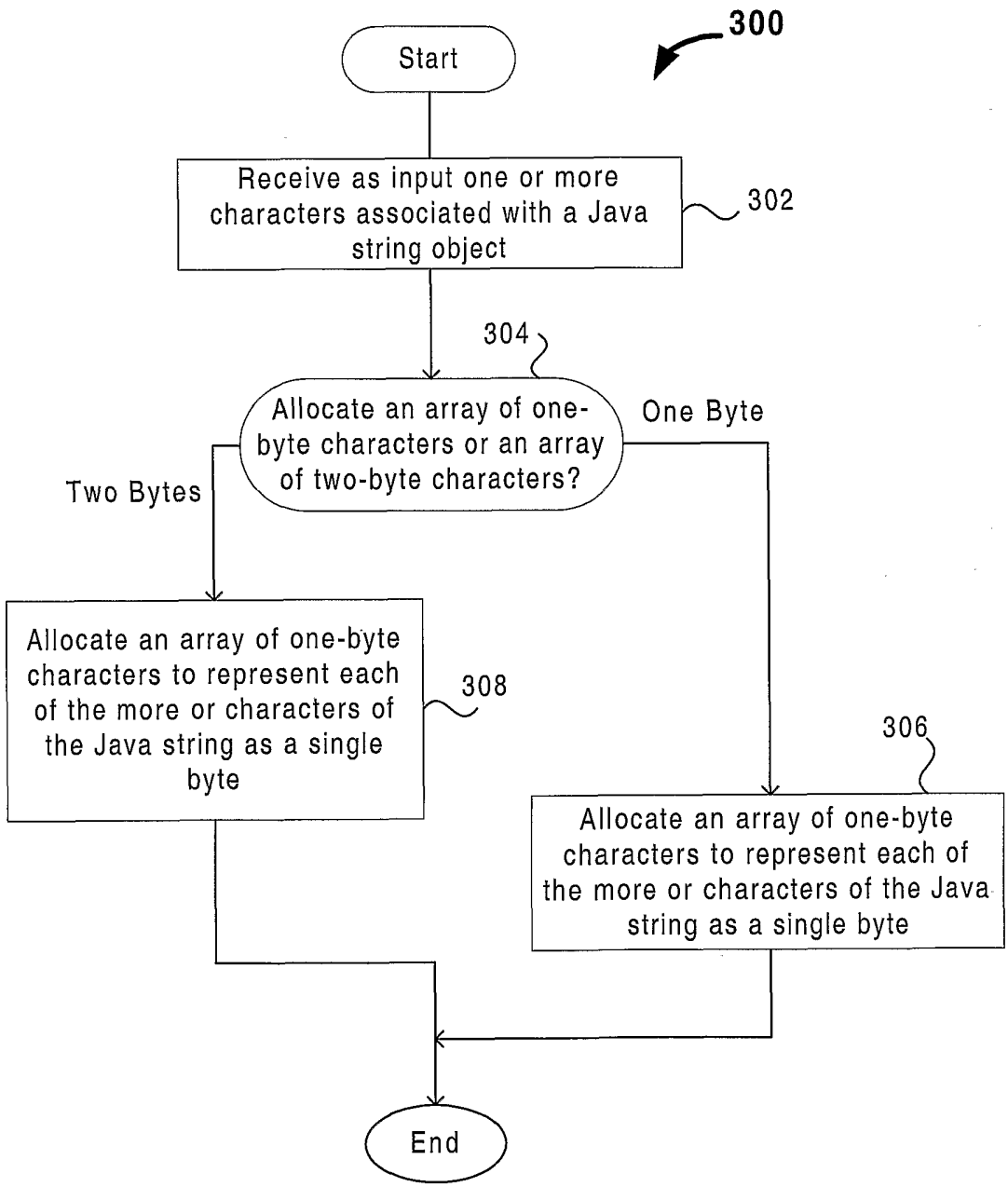

300
Start
Receive as input one or more characters associated with a Java string object
302
304
Allocate an array of one-byte characters or an array of two-byte characters?
One Byte
Two Bytes
Allocate an array of one-byte characters to represent each of the more or characters of the Java string as a single byte
308
306
Allocate an array of one-byte characters to represent each of the more or characters of the Java string as a single byte
End

Fig. 3

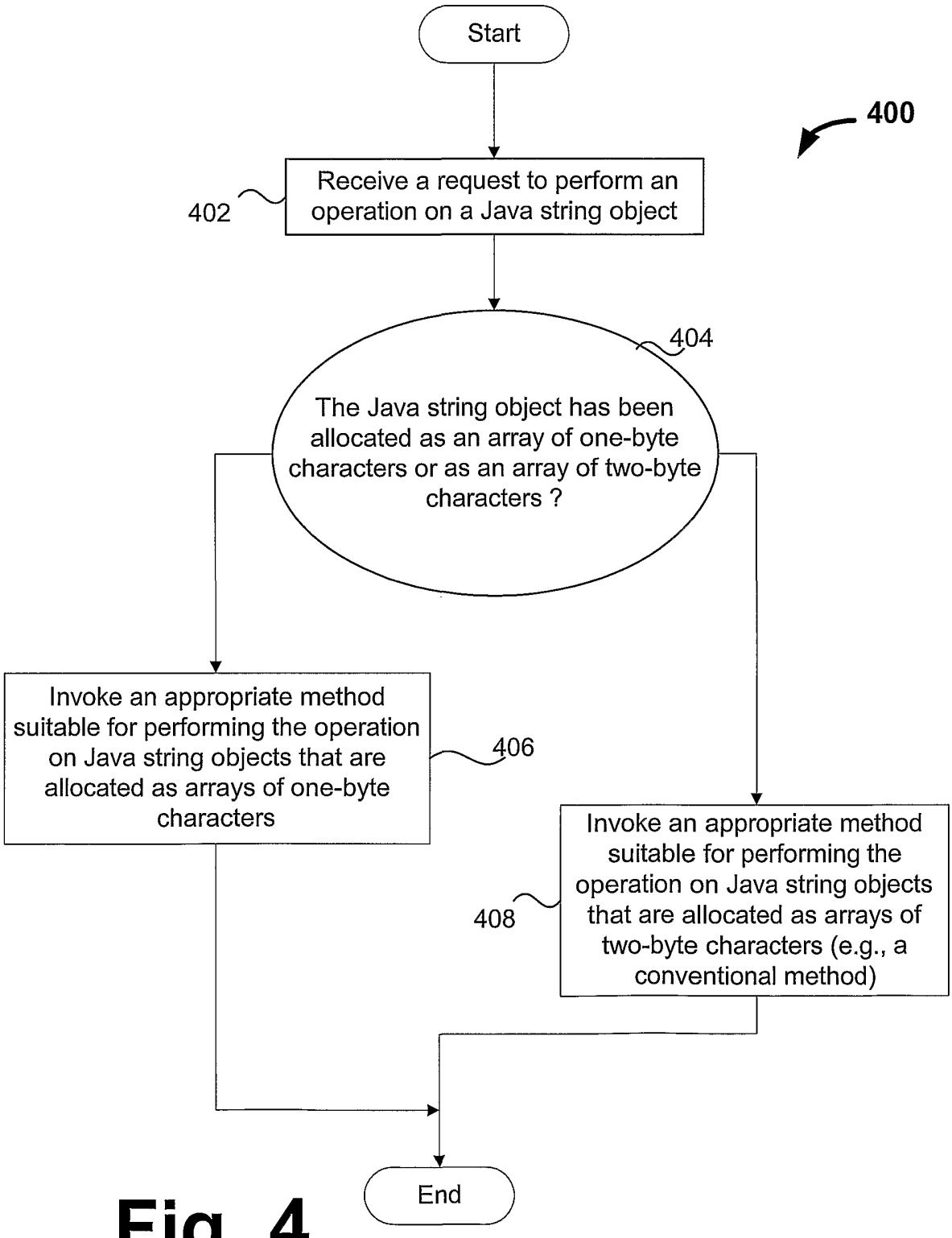

Start
400
Receive a request to perform an operation on a Java string object
402
404
The Java string object has been allocated as an array of one-byte characters or as an array of two-byte characters ?
Invoke an appropriate method suitable for performing the operation on Java string objects that are allocated as arrays of one-byte characters
406
Invoke an appropriate method suitable for performing the operation on Java string objects that are allocated as arrays of two-byte characters (e.g., a conventional method)
408
End

Fig. 4