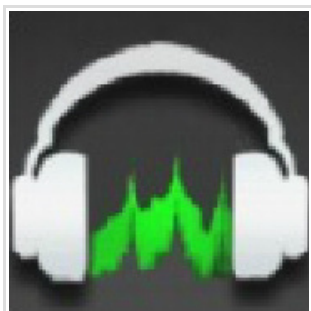


# [conowen]大钟的专栏

[目录视图](#)[摘要视图](#)[RSS 订阅](#)

个人资料



conowen



访问: 782170次

积分: 7778

等级:

## Android的NDK开发(3)——JNI数据类型的详解

标签: [jni](#) [android](#) [reference](#) [primitive](#) [java](#)

2012-04-30 19:02

20557人阅读

[评论\(4\)](#)[收藏](#)[举报](#)[分类:](#) [Android的NDK开发 \(4\)](#) ▲

版权声明: 本文为博主原创文章, 未经博主允许不得转载。

/\*\*\*\*\*

\* author: conowen@大钟

\* E-mail: conowen@hotmail.com

\*site:http://menwoo.com/

\*深圳市大望谷科技有限公司

排名: 第1309名

原创: 58篇

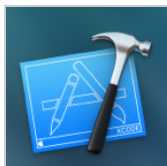
转载: 13篇

译文: 0篇

评论: 430条

文章搜索

博客专栏



大钟的ios开发之旅

文章: 4篇

阅读: 13443



大钟的  
Android\_NDK开发

文章: 5篇

阅读: 101611



Android学习笔  
记

文章: 24篇

阅读: 289827

文章分类

android编译 (8)

Android学习笔记 (24)

\* <http://blog.csdn.net/conowen>

\* 注: 本文为原创, 仅作为学习交流使用, 转载请标明作者及出处。

\*\*\*\*\*/

在Java中有两类数据类型: **primitive types**, 如, `int`, `float`, `char`; 另一种为 **reference types**, 如, 类, 实例, 数组。

**注意:** 数组, 不管是对象数组还是基本类型数组, 都作为**reference types**存在, 有专门的JNI方法取数组中每个元素。

## 1、void

java的void与JNI的void是一致的。

## 2、基本数据类型

- JAVA学习 (1)
- C与C++ (2)
- Linux相关 (4)
- Android相关 (8)
- 瑞芯微RK (3)
- 嵌入式相关 (1)
- 计算机相关 (2)
- Android多媒体&流媒体开发 (6)
- Android的NDK开发 (5)
- 所想所感 (1)
- 所想所感 ios (1)

文章存档

- 2015年09月 (1)
- 2014年11月 (4)
- 2014年04月 (2)
- 2012年08月 (6)
- 2012年07月 (4)

展开

阅读排行

- Android学习笔记(21)——  
【整理】Android-Recover (50913)
- Android的NDK开发(1)—— (45982)
- Android学习笔记(13)—— (34034)
- Android学习笔记(12)—— (26907)
- Android图形系统之Surface (22787)

	A	B	C	D
1	基本数据类型			
2	Java类型	JNI 类型	纯C/C++类型	描述
3	boolean	jboolean	unsigned char 或 uint8_t	C/C++8位整型
4	byte	jbyte	signed char 或 int8_t	C/C++有符号的8位整型
5	char	jchar	unsigned short 或 uint16_t	C/C++无符号的16位整型
6	short	jshort	short 或 int16_t	C/C++有符号的16位整型
7	int	jint	int 或 int32_t	C/C++有符号的32位整型
8	long	jlong	long long 或 int64_t	C/C++有符号的64位整型
9	float	jfloat	float @conowen	C/C++32位浮点型
10	double	jdouble	double	C/C++64位浮点型

3、对象类型

【整理】Libav、FFmpeg

(22168)

Android的文件系统结构

(21748)

RKAndroidTool工具的各

(21537)

Android的NDK开发(3)——

(21504)

(20550)

评论排行

Android学习笔记(21)——

(76)

关于havenapetr-FFMpeg

(63)

Android多媒体开发（3）

(51)

Android多媒体开发（4）

(46)

Android多媒体开发（5）

(41)

Android学习笔记(20)——

(15)

Android学习笔记(12)——

(14)

Android多媒体开发（2）

(14)

Android学习笔记(1)——

(9)

Android的NDK开发(2)——

(8)

推荐文章

\*在R中使用支持向量机（SVM）  
进行数据挖掘（上）

\* 你不再需要动态网页——编辑-  
发布-开发分离

\*Android性能优化之使用线程池  
处理异步任务

\*通过Ajax的方式执行GP服务

\*编译器架构的王者LLVM——

	A	B	C
1	对象类型		
2	Java类型	JNI 类型	描述
3	Object	jobject	任意Java对象，或者没有对应java类型的对象
4	Class	jclass	Class对象
5	String	jstring	字符串对象 @conowen
6			

相比基本类型，对象类型的传递要复杂得多。不能对Jstring进行直接操作。

[java]

C

?

```
01. //如下使用方式是错误的，因为jstring不同于C语言中的char *类型。
02. Java_com_conowen_test_testActivity_test(JNIEnv *env, jobject obj, jstring str)
03. {
04.     /* ERROR: incorrect use of jstring as a char* pointer */
05.     printf("%s", str);
06.     ...
07. }
```

注意：

(6) 多遍翻译的宏翻译系统

\* 【笨木头Unity】入门之旅

010 (完结): Demo之四处找死

(五) \_UI

最新评论

- Android多媒体开发 (3) ——勤修戒定慧: Hi,哥们儿。libjniaudio.so和libjnivideo.so你是在linux操作系统中通...
- Android学习笔记(21)——利风之字符: @jicancheng:大神 可以加我Q648131109吗 这个问题我卡了好久
- Android学习笔记(21)——利风之字符: com.mysql.jdbc.CommunicationsException: Communicat...
- Android多媒体开发 (5) ——guonanyun: 楼主,可以提供一下完整demo的下载地址吗?
- Android的NDK开发(4)——J wangjg0317: 写得很好, 谢谢楼主, 楼主能否写个http相关的例子?
- Android多媒体开发 (5) ——u014453626: @zhf198909:你的问题我也出现过, 经过反复播放视频, 发现博主在onCreate中就已经放入M...
- Android多媒体开发 (5) ——u014453626: @zhf198909:你的问题我也出现过, 经过反复播放视频, 发现博主在onCreate中就已经放入M...
- 【整理】SISD、MIMD、SIMD、I mayfla: 非常棒, 学习了
- Android学习笔记(21)——利sanbo\_xyz

[java] C 8

01. typedef jint jsize;

A		B
JNI String 相关函数		描述
3		
4	const jchar * GetStringChars(JNIEnv *env, jstring str, jboolean *isCopy);	得到Unicode编码的String指针, 返回值为string的copy值
5	void ReleaseStringChars(JNIEnv *env, jstring string, const jchar *chars);	释放
6	const jbyte * GetStringUTFChars(JNIEnv *env, jstring string, jboolean *isCopy);	得到UTF-8编码的String指针, 返回值为string的copy值
7	void ReleaseStringUTFChars(JNIEnv *env, jstring string, const char *utf);	释放
8	jsize GetStringLength(JNIEnv *env, jstring string);	得到Unicode编码格式的String长度
9	jsize GetStringUTFLength(JNIEnv *env, jstring string);	得到UTF-8编码格式的String长度
10	jstring NewString(JNIEnv *env, const jchar *uchars, jsize len);	创建一个java.lang.String实例, 长度与参数中给出的Unicode编码格式的String相同
11	jstring NewStringUTF(JNIEnv *env, const char *bytes);	创建一个java.lang.String实例, 长度与参数中给出的UTF-8编码格式的String相同
12	const jchar * GetStringCritical(JNIEnv *env, jstring string, jboolean *isCopy);	得到编码格式的String的指针, 返回String的copy值
13	void ReleaseStringCritical(JNIEnv *env, jstring string, const jchar *carray);	释放 @conowen
14	void GetStringRegion(JNIEnv *env, jstring str, jsize start, jsize len, jchar *buf); (*env)->SetStringRegion	把String的内容复制出来, 复制给Unicode编码格式的参数buf
15	void GetStringUTFRegion(JNIEnv *env, jstring str, jsize start, jsize len, jchar *buf) (*env)->SetStringUTFRegion	把String的内容复制出来, 复制给UTF-8编码格式的参数buf

3.1、GetStringUTFChars与ReleaseStringUTFChars函数简单说明 (跳到3.2有更方便的函数)

JNI支持Unicode/UTF-8字符编码互转。Unicode以16-bits值编码; UTF-8是一种以字节为单位变长格式的字符编码, 并与7-bitsASCII码兼容。UTF-8字串与C字串一样, 以NULL('\0')做结束符, 当UTF-8包含非ASCII码字符

```
java.lang.ClassNotFoundException:
org.gjt.mm.mysql...
```

Android学习笔记(13)——利  
wlccomeon: 使用的挺方便的,  
现在忘了, 来看看。

时, 以'\0'做结束符的规则不变。7-bit ASCII字符的取值范围在1-127之间, 这些字符的值域与UTF-8中相同。当最高位被设置时, 表示多字节编码。

```
[java] C ?

01. //调用GetStringUTFChars, 把一个Unicode字符串转成UTF-8格式字符串
02.
03. Java_com_conowen_test_testActivity_test(JNIEnv *env, jobject obj, jstring str)
04. {
05.     char buf[128];
06.     const jbyte *cbyte;
07.     cbyte= (*env)->GetStringUTFChars(env, str, NULL);
08.     if (cbyte== NULL) {
09.         return NULL;
10.     }
11.     printf("%s", cbyte);
12.     (*env)->ReleaseStringUTFChars(env, str, cbyte);
13.
14.     scanf("%127s", buf);
15.     return (*env)->NewStringUTF(env, buf);
16.
17. //或者return (*env)->NewStringUTF(env, "hello world");
18. }
```

上述函数中, 有isCopy参数, 当该值为JNI\_TRUE, 将返回str的一个拷贝; 为JNI\_FALSE将直接指向str的内容。注意: 当isCopy为JNI\_FALSE, 不要修改返回值, 不然将改变java.lang.String的不可变语义。一般会把isCopy设为NULL, 不关心Java VM对返回的指针是否直接指向java.lang.String的内容。

注意: 在调用GetStringChars之后, 一定要调用ReleaseStringChars做释放, (Unicode -> UTF-8转换的原因)。不管在调用GetStringChars时为isCopy赋值JNI\_TRUE还是JNI\_FALSE, 因不同JavaVM实现的原因, ReleaseStringChars可能释放内存, 也可能释放一个内存占用标记。

### 3.2、GetStringRegion/GetStringUTFRegion函数简单说明

因为这两个函数不涉及内存操作，所以较GetStringUTFChars使用要简单。也不用进行释放指针之类的操作，非常方便。（推荐使用）

```
[java] C ?
01.  Java_com_conowen_test_testActivity_test(JNIEnv *env, jobject obj, jstring str)
02.  {
03.
04.      char outputbuf[128], inputbuf[128];
05.      int len = (*env)->GetStringLength(env, str);
06.      (*env)->GetStringUTFRegion(env, str, 0, len, outputbuf);
07.      printf("%s", outputbuf);
08.      scanf("%s", inputbuf);
09.      return (*env)->NewStringUTF(env, inbuf);
10. }
```

GetStringUTFRegion有两个主要的参数，start 和 length, 这两个参数以Unicode编码计算. 该函数会做边界检查，所以可能抛出StringIndexOutOfBoundsException。

### 3.3、GetStringLength/GetStringUTFLength函数简单说明

前者是Unicode编码长度，后者返回的是UTF编码长度。

## 4、数组类型



	A	B	C
1	JNI数组类型		
2	Java类型	JNI 类型	描述
3	boolean[]	jbooleanArray	布尔型数组
4	byte[]	jbyteArray	比特型数组
5	char[]	jcharArray	字符型数组
6	short[]	jshortArray	短整型数组
7	int[]	jintArray	整型数组 @conowen
8	long[]	jlongArray	长整型数组
9	float[]	jfloatArray	浮点型数组
10	double[]	jdoubleArray	双浮点型数组

JNI对每种数据类型的数组都有对应的函数。

4.1、常见错误操作：

[java]

C

8

01. /\* 直接操作数组是错误的 \*/



```
02.   Java_IntArray_sumArray(JNIEnv *env, jobject obj, jintArray arr)
03.   {
04.       int i, sum = 0;
05.       for (i = 0; i < 10; i++) {
06.           sum += arr[i];
07.       }
08.   }
```

## 4.2、使用

`void Get<Type>ArrayRegion(JNIEnv *env,<ArrayType> array, jsize start,jsize len,  
<NativeType> *buf);`

进行操作

参数说明:

**env**: the JNIEnv interface pointer.

**array**: a reference to an array whose elements are to be copied.将要被拷贝的目标数组<ArrayType>

**start**: the starting index of the array elements to be copied. (数组的起始位置)

**len**: the number of elements to be copied. (拷贝元素的个数) **buf**:the destination buffer.存放结果的本地数组  
<NativeType>

返回值: void

<i>Get&lt;Type&gt;ArrayRegion</i>	<i>&lt;ArrayType&gt;</i>	<i>&lt;NativeType&gt;</i>
GetBooleanArrayRegion	jbooleanArray	jboolean
GetByteArrayRegion	jbyteArray	jbyte
GetCharArrayRegion	jcharArray	jchar
GetShortArrayRegion	jshortArray	jhort
GetIntArrayRegion	jintArray	jint
GetLongArrayRegion	jlongArray	jlong
GetFloatArrayRegion	jfloatArray	jfloat
GetDoubleArrayRegion	jdoubleArray	jdouble

[java] C }

```
01. Java_IntArray_sumArray(JNIEnv *env, jobject obj, jintArray arr)
02. {
03.     jint buf[10];
04.     jint i, sum = 0;
05.     (*env)->GetIntArrayRegion(env, arr, 0, 10, buf);
06.     for (i = 0; i < 10; i++) {
07.         sum += buf[i];
08.     }
09.     return sum;
10. }
11.
12. JNI中数组的基类为jarray，其他如jintArray都是继承自jarray。
```

4.3、使用<NativeType> \*Get<Type>ArrayElements(JNIEnv \*env,<ArrayType> array, jboolean \*isCopy);进行数组操作

<i>Get&lt;Type&gt;ArrayElements</i>	<i>&lt;ArrayType&gt;</i>	<i>&lt;NativeType&gt;</i>
GetBooleanArrayElements	jbooleanArray	jboolean
GetByteArrayElements	jbyteArray	jbyte
GetCharArrayElements	jcharArray	jchar
GetShortArrayElements	jshortArray	jshort
GetIntArrayElements	jintArray	jint
GetLongArrayElements	jlongArray	jlong
GetFloatArrayElements	jfloatArray	jfloat
GetDoubleArrayElements	jdoubleArray	jdouble

参数说明:

**env:** the JNIEnv interface pointer.array: a reference to the primitive array whose elements are to be accessed.  
(目标数组)

**isCopy:** a pointer to a jboolean indicating whether a function

返回值: 返回指向Java数组的一个直接的指针

[java]

```
01. 使用实例:
02.
03. Java_IntArray_sumArray(JNIEnv *env, jobject obj, jintArray arr)
04. {
```

```
05.     jint *carr;  
06.     jint i, sum = 0;  
07.     carr = (*env)->GetIntArrayElements(env, arr, NULL);  
08.     if (carr == NULL) {  
09.         return 0; /* exception occurred */  
10.     }  
11.     for (i=0; i<10; i++) {  
12.         sum += carr[i];  
13.     }  
14.     (*env)->ReleaseIntArrayElements(env, arr, carr, 0);  
15.     return sum;  
16. }  
17.  
18.  
19. )
```

更多数组操作函数:

JNI Function	Description	Since
<i>Get&lt;Type&gt;ArrayRegion</i> <i>Set&lt;Type&gt;ArrayRegion</i>	Copies the contents of primitive arrays to or from a preallocated C buffer.	JDK1.1
<i>Get&lt;Type&gt;ArrayElements</i> <i>Release&lt;Type&gt;ArrayElements</i>	Obtains a pointer to the contents of a primitive array. May return a copy of the array.	JDK1.1
<i>GetArrayLength</i>	Returns the number of elements in the array.	JDK1.1
<i>New&lt;Type&gt;Array</i>	Creates an array with the given length.	JDK1.1
<i>GetPrimitiveArrayCritical</i> <i>ReleasePrimitiveArrayCritical</i>	Obtains or releases a pointer to the contents of a primitive array. May disable garbage collection, or return a copy of the array.	Java 2 SDK1.2

5、另外一些有用的宏定义（来自jni.h）

[java]C⌘

```
01. #define JNI_FALSE 0
02. #define JNI_TRUE 1
03.
04. #define JNI_VERSION_1_1 0x00010001
05. #define JNI_VERSION_1_2 0x00010002
06. #define JNI_VERSION_1_4 0x00010004
07. #define JNI_VERSION_1_6 0x00010006
08.
09. #define JNI_OK (0) /* no error */
10. #define JNI_ERR (-1) /* generic error */
11. #define JNI_EDETACHED (-2) /* thread detached from the VM */
```

```
12.  #define JNI_EVERSION      (-3)          /* JNI version error */
13.
14.  #define JNI_COMMIT         1             /* copy content, do not free buffer */
15.  #define JNI_ABORT          2             /* free buffer w/o copying back */
```

上一篇
 Android的NDK开发(2)——利用Android NDK编写一个简单的HelloWorld
 下一篇
 Android的NDK开发(4)——JNI数据结构之JNINativeMethod

顶
 踩
 0
 0

我的同类文章

Android的NDK开发（4）

• Android的NDK开发(5)——Android JNI层实...
 • Android的NDK开发(4)——JNI数据结构之JNI...
 • Android的NDK开发(2)——利用Android NDK...
 • Android的NDK开发(1)——Android JNI简介...

主题推荐
 android
 color
 数据
 ndk
 jni
 class
 rgb

猜你在找

Android底层技术：Java层系统服务(Android Service) Android的NDK开发3JNI数据类型的详解



[ArcGIS for JavaScript](#)[Android的NDK开发3JNI数据类型的详解](#)[Java Swing、JDBC开发桌面级应用](#)[Android的NDK开发3JNI数据类型的详解](#)[Android必备的Java基础知识\(三\)](#)[Android的NDK开发3JNI数据类型的详解](#)[\[oeasy\]教你玩转java编程-我的世界mc编程入门](#)[Android的NDK开发3JNI数据类型的详解](#)[查看评论](#)3楼 [nanrenzhenghao](#) 2013-12-19 14:46发表

感谢楼组分享！调用成功了！

2楼 [maggiccrystal\\_3](#) 2013-04-03 11:08发表

楼主，你好，我想请教一下：`jstring getName(JNIEnv *env, jobject thiz){  
return env->NewStringUTF("CCTV1");  
}`

如果循环调用这个方法`env->NewStringUTF()` 这个返回的string 我是在Java层得到的string name=null 还是JNI这里释放？

1楼 [tx3344](#) 2012-09-15 23:20发表

点点滴滴

Re: [程序猿工会主席](#) 2013-12-25 17:03发表

回复tx3344：在JNI里释放。

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

\* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

[全部主题](#)[Hadoop](#)[AWS](#)[移动游戏](#)[Java](#)[Android](#)[iOS](#)[Swift](#)[智能硬件](#)[Docker](#)[OpenStack](#)[VPN](#)[Spark](#)[ERP](#)[IE10](#)[Eclipse](#)[CRM](#)[JavaScript](#)[数据库](#)[Ubuntu](#)[NFC](#)[WAP](#)[jQuery](#)

[BI](#) [HTML5](#) [Spring](#) [Apache](#) [.NET](#) [API](#) [HTML](#) [SDK](#) [IIS](#) [Fedora](#) [XML](#) [LBS](#) [Unity](#)  
[Splashtop](#) [UML](#) [components](#) [Windows Mobile](#) [Rails](#) [QEMU](#) [KDE](#) [Cassandra](#) [CloudStack](#)  
[FTC](#) [coremail](#) [OPhone](#) [CouchBase](#) [云计算](#) [iOS6](#) [Rackspace](#) [Web App](#) [SpringSide](#)  
[Maemo](#) [Compuware](#) [大数据](#) [aptech](#) [Perl](#) [Tornado](#) [Ruby](#) [Hibernate](#) [ThinkPHP](#) [HBase](#)  
[Pure](#) [Solr](#) [Angular](#) [Cloud Foundry](#) [Redis](#) [Scala](#) [Django](#) [Bootstrap](#)

[公司简介](#) | [招贤纳士](#) | [广告服务](#) | [银行汇款帐号](#) | [联系方式](#) | [版权声明](#) | [法律顾问](#) | [问题报告](#) | [合作伙伴](#) | [论坛反馈](#)

[网站客服](#)   [杂志客服](#)   [微博客服](#)   [webmaster@csdn.net](mailto:webmaster@csdn.net)   400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏乐知网络技术有限公司 提供商务支持  
京 ICP 证 070598 号 | Copyright © 1999-2014, CSDN.NET, All Rights Reserved 