| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ArrayType.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PREV CLASS   [**NEXT CLASS**](http://docs.google.com/javax/management/openmbean/CompositeData.html) | [**FRAMES**](http://docs.google.com/index.html?javax/management/openmbean/ArrayType.html)    [**NO FRAMES**](http://docs.google.com/ArrayType.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#2et92p0) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: FIELD | [CONSTR](#2s8eyo1) | [METHOD](#26in1rg) |

## **javax.management.openmbean**

Class ArrayType<T>

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [javax.management.openmbean.OpenType](http://docs.google.com/javax/management/openmbean/OpenType.html)<T>  
 **javax.management.openmbean.ArrayType<T>**

**All Implemented Interfaces:** [Serializable](http://docs.google.com/java/io/Serializable.html)

public class **ArrayType<T>**extends [OpenType](http://docs.google.com/javax/management/openmbean/OpenType.html)<T>

The ArrayType class is the *open type* class whose instances describe all *open data* values which are n-dimensional arrays of *open data* values.

Examples of valid ArrayType instances are:

// 2-dimension array of java.lang.String  
 ArrayType a1 = new ArrayType(2, SimpleType.STRING);  
  
 // 1-dimension array of int  
 ArrayType a2 = new ArrayType(SimpleType.INTEGER, true);  
  
 // 1-dimension array of java.lang.Integer  
 ArrayType a3 = new ArrayType(SimpleType.INTEGER, false);  
  
 // 4-dimension array of int  
 ArrayType a4 = new ArrayType(3, a2);  
  
 // 4-dimension array of java.lang.Integer  
 ArrayType a5 = new ArrayType(3, a3);  
  
 // 1-dimension array of java.lang.String  
 ArrayType a6 = new ArrayType(SimpleType.STRING, false);  
  
 // 1-dimension array of long  
 ArrayType a7 = new ArrayType(SimpleType.LONG, true);  
  
 // 1-dimension array of java.lang.Integer  
 ArrayType a8 = ArrayType.getArrayType(SimpleType.INTEGER);  
  
 // 2-dimension array of java.lang.Integer  
 ArrayType a9 = ArrayType.getArrayType(a8);  
  
 // 2-dimension array of int  
 ArrayType a10 = ArrayType.getPrimitiveArrayType(int[][].class);  
  
 // 3-dimension array of int  
 ArrayType a11 = ArrayType.getArrayType(a10);  
  
 // 1-dimension array of float  
 ArrayType a12 = ArrayType.getPrimitiveArrayType(float[].class);  
  
 // 2-dimension array of float  
 ArrayType a13 = ArrayType.getArrayType(a12);  
  
 // 1-dimension array of javax.management.ObjectName  
 ArrayType a14 = ArrayType.getArrayType(SimpleType.OBJECTNAME);  
  
 // 2-dimension array of javax.management.ObjectName  
 ArrayType a15 = ArrayType.getArrayType(a14);  
  
 // 3-dimension array of java.lang.String  
 ArrayType a16 = new ArrayType(3, SimpleType.STRING);  
  
 // 1-dimension array of java.lang.String  
 ArrayType a17 = new ArrayType(1, SimpleType.STRING);  
  
 // 2-dimension array of java.lang.String  
 ArrayType a18 = new ArrayType(1, a17);  
  
 // 3-dimension array of java.lang.String  
 ArrayType a19 = new ArrayType(1, a18);

**Since:** 1.5 **See Also:**[Serialized Form](http://docs.google.com/serialized-form.html#javax.management.openmbean.ArrayType)

| **Field Summary** | |
| --- | --- |

| **Fields inherited from class javax.management.openmbean.**[**OpenType**](http://docs.google.com/javax/management/openmbean/OpenType.html) |
| --- |
| [ALLOWED\_CLASSNAMES](http://docs.google.com/javax/management/openmbean/OpenType.html#ALLOWED_CLASSNAMES), [ALLOWED\_CLASSNAMES\_LIST](http://docs.google.com/javax/management/openmbean/OpenType.html#ALLOWED_CLASSNAMES_LIST) |

| **Constructor Summary** | |
| --- | --- |
| [**ArrayType**](http://docs.google.com/javax/management/openmbean/ArrayType.html#ArrayType(int,%20javax.management.openmbean.OpenType))(int dimension, [OpenType](http://docs.google.com/javax/management/openmbean/OpenType.html)<?> elementType)            Constructs an ArrayType instance describing *open data* values which are arrays with dimension dimension of elements whose *open type* is elementType. |
| [**ArrayType**](http://docs.google.com/javax/management/openmbean/ArrayType.html#ArrayType(javax.management.openmbean.SimpleType,%20boolean))([SimpleType](http://docs.google.com/javax/management/openmbean/SimpleType.html)<?> elementType, boolean primitiveArray)            Constructs a unidimensional ArrayType instance for the supplied SimpleType. |

| **Method Summary** | |
| --- | --- |
| boolean | [**equals**](http://docs.google.com/javax/management/openmbean/ArrayType.html#equals(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) obj)            Compares the specified obj parameter with this ArrayType instance for equality. |
| static   | <E> [ArrayType](http://docs.google.com/javax/management/openmbean/ArrayType.html)<E[]> | | --- | | [**getArrayType**](http://docs.google.com/javax/management/openmbean/ArrayType.html#getArrayType(javax.management.openmbean.OpenType))([OpenType](http://docs.google.com/javax/management/openmbean/OpenType.html)<E> elementType)            Create an ArrayType instance in a type-safe manner. |
| int | [**getDimension**](http://docs.google.com/javax/management/openmbean/ArrayType.html#getDimension())()            Returns the dimension of arrays described by this ArrayType instance. |
| [OpenType](http://docs.google.com/javax/management/openmbean/OpenType.html)<?> | [**getElementOpenType**](http://docs.google.com/javax/management/openmbean/ArrayType.html#getElementOpenType())()            Returns the *open type* of element values contained in the arrays described by this ArrayType instance. |
| static   | <T> [ArrayType](http://docs.google.com/javax/management/openmbean/ArrayType.html)<T> | | --- | | [**getPrimitiveArrayType**](http://docs.google.com/javax/management/openmbean/ArrayType.html#getPrimitiveArrayType(java.lang.Class))([Class](http://docs.google.com/java/lang/Class.html)<T> arrayClass)            Create an ArrayType instance in a type-safe manner. |
| int | [**hashCode**](http://docs.google.com/javax/management/openmbean/ArrayType.html#hashCode())()            Returns the hash code value for this ArrayType instance. |
| boolean | [**isPrimitiveArray**](http://docs.google.com/javax/management/openmbean/ArrayType.html#isPrimitiveArray())()            Returns true if the open data values this open type describes are primitive arrays, false otherwise. |
| boolean | [**isValue**](http://docs.google.com/javax/management/openmbean/ArrayType.html#isValue(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) obj)            Tests whether obj is a value for this ArrayType instance. |
| [String](http://docs.google.com/java/lang/String.html) | [**toString**](http://docs.google.com/javax/management/openmbean/ArrayType.html#toString())()            Returns a string representation of this ArrayType instance. |

| **Methods inherited from class javax.management.openmbean.**[**OpenType**](http://docs.google.com/javax/management/openmbean/OpenType.html) |
| --- |
| [getClassName](http://docs.google.com/javax/management/openmbean/OpenType.html#getClassName()), [getDescription](http://docs.google.com/javax/management/openmbean/OpenType.html#getDescription()), [getTypeName](http://docs.google.com/javax/management/openmbean/OpenType.html#getTypeName()), [isArray](http://docs.google.com/javax/management/openmbean/OpenType.html#isArray()) |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [clone](http://docs.google.com/java/lang/Object.html#clone()), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

| **Constructor Detail** |
| --- |

### ArrayType

public **ArrayType**(int dimension,  
 [OpenType](http://docs.google.com/javax/management/openmbean/OpenType.html)<?> elementType)  
 throws [OpenDataException](http://docs.google.com/javax/management/openmbean/OpenDataException.html)

Constructs an ArrayType instance describing *open data* values which are arrays with dimension dimension of elements whose *open type* is elementType.

When invoked on an ArrayType instance, the [getClassName](http://docs.google.com/javax/management/openmbean/OpenType.html#getClassName()) method returns the class name of the array instances it describes (following the rules defined by the [getName](http://docs.google.com/java/lang/Class.html#getName()) method of java.lang.Class), not the class name of the array elements (which is returned by a call to getElementOpenType().getClassName()).

The internal field corresponding to the type name of this ArrayType instance is also set to the class name of the array instances it describes. In other words, the methods getClassName and getTypeName return the same string value. The internal field corresponding to the description of this ArrayType instance is set to a string value which follows the following template:

* if non-primitive array: *<dimension>*-dimension array of *<element\_class\_name>*
* if primitive array: *<dimension>*-dimension array of *<primitive\_type\_of\_the\_element\_class\_name>*

As an example, the following piece of code:

ArrayType t = new ArrayType(3, SimpleType.STRING);  
 System.out.println("array class name = " + t.getClassName());  
 System.out.println("element class name = " + t.getElementOpenType().getClassName());  
 System.out.println("array type name = " + t.getTypeName());  
 System.out.println("array type description = " + t.getDescription());

would produce the following output:

array class name = [[[Ljava.lang.String;  
 element class name = java.lang.String  
 array type name = [[[Ljava.lang.String;  
 array type description = 3-dimension array of java.lang.String

And the following piece of code which is equivalent to the one listed above would also produce the same output:

ArrayType t1 = new ArrayType(1, SimpleType.STRING);  
 ArrayType t2 = new ArrayType(1, t1);  
 ArrayType t3 = new ArrayType(1, t2);  
 System.out.println("array class name = " + t3.getClassName());  
 System.out.println("element class name = " + t3.getElementOpenType().getClassName());  
 System.out.println("array type name = " + t3.getTypeName());  
 System.out.println("array type description = " + t3.getDescription());

**Parameters:**dimension - the dimension of arrays described by this ArrayType instance; must be greater than or equal to 1.elementType - the *open type* of element values contained in the arrays described by this ArrayType instance; must be an instance of either SimpleType, CompositeType, TabularType or another ArrayType with a SimpleType, CompositeType or TabularType as its elementType. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if dimension is not a positive integer. [OpenDataException](http://docs.google.com/javax/management/openmbean/OpenDataException.html) - if elementType's className is not one of the allowed Java class names for open data.

### ArrayType

public **ArrayType**([SimpleType](http://docs.google.com/javax/management/openmbean/SimpleType.html)<?> elementType,  
 boolean primitiveArray)  
 throws [OpenDataException](http://docs.google.com/javax/management/openmbean/OpenDataException.html)

Constructs a unidimensional ArrayType instance for the supplied SimpleType.

This constructor supports the creation of arrays of primitive types when primitiveArray is true.

For primitive arrays the [getElementOpenType()](http://docs.google.com/javax/management/openmbean/ArrayType.html#getElementOpenType()) method returns the [SimpleType](http://docs.google.com/javax/management/openmbean/SimpleType.html) corresponding to the wrapper type of the primitive type of the array.

When invoked on an ArrayType instance, the [getClassName](http://docs.google.com/javax/management/openmbean/OpenType.html#getClassName()) method returns the class name of the array instances it describes (following the rules defined by the [getName](http://docs.google.com/java/lang/Class.html#getName()) method of java.lang.Class), not the class name of the array elements (which is returned by a call to getElementOpenType().getClassName()).

The internal field corresponding to the type name of this ArrayType instance is also set to the class name of the array instances it describes. In other words, the methods getClassName and getTypeName return the same string value. The internal field corresponding to the description of this ArrayType instance is set to a string value which follows the following template:

* if non-primitive array: 1-dimension array of *<element\_class\_name>*
* if primitive array: 1-dimension array of *<primitive\_type\_of\_the\_element\_class\_name>*

As an example, the following piece of code:

ArrayType t = new ArrayType(SimpleType.INTEGER, true);  
 System.out.println("array class name = " + t.getClassName());  
 System.out.println("element class name = " + t.getElementOpenType().getClassName());  
 System.out.println("array type name = " + t.getTypeName());  
 System.out.println("array type description = " + t.getDescription());

would produce the following output:

array class name = [I  
 element class name = java.lang.Integer  
 array type name = [I  
 array type description = 1-dimension array of int

**Parameters:**elementType - the SimpleType of the element values contained in the arrays described by this ArrayType instance.primitiveArray - true when this array describes primitive arrays. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if dimension is not a positive integer. [OpenDataException](http://docs.google.com/javax/management/openmbean/OpenDataException.html) - if primitiveArray is true and elementType is not a valid SimpleType for a primitive type.**Since:** 1.6

| **Method Detail** |
| --- |

### getDimension

public int **getDimension**()

Returns the dimension of arrays described by this ArrayType instance.

**Returns:**the dimension.

### getElementOpenType

public [OpenType](http://docs.google.com/javax/management/openmbean/OpenType.html)<?> **getElementOpenType**()

Returns the *open type* of element values contained in the arrays described by this ArrayType instance.

**Returns:**the element type.

### isPrimitiveArray

public boolean **isPrimitiveArray**()

Returns true if the open data values this open type describes are primitive arrays, false otherwise.

**Returns:**true if this is a primitive array type.**Since:** 1.6

### isValue

public boolean **isValue**([Object](http://docs.google.com/java/lang/Object.html) obj)

Tests whether obj is a value for this ArrayType instance.

This method returns true if and only if obj is not null, obj is an array and any one of the following is true:

* if this ArrayType instance describes an array of SimpleType elements or their corresponding primitive types, obj's class name is the same as the className field defined for this ArrayType instance (i.e. the class name returned by the [getClassName](http://docs.google.com/javax/management/openmbean/OpenType.html#getClassName()) method, which includes the dimension information),
* if this ArrayType instance describes an array of classes implementing the TabularData interface or the CompositeData interface, obj is assignable to such a declared array, and each element contained in obj is either null or a valid value for the element's open type specified by this ArrayType instance.

**Specified by:**[isValue](http://docs.google.com/javax/management/openmbean/OpenType.html#isValue(java.lang.Object)) in class [OpenType](http://docs.google.com/javax/management/openmbean/OpenType.html)<[T](http://docs.google.com/javax/management/openmbean/ArrayType.html)> **Parameters:**obj - the object to be tested. **Returns:**true if obj is a value for this ArrayType instance.

### equals

public boolean **equals**([Object](http://docs.google.com/java/lang/Object.html) obj)

Compares the specified obj parameter with this ArrayType instance for equality.

Two ArrayType instances are equal if and only if they describe array instances which have the same dimension, elements' open type and primitive array flag.

**Specified by:**[equals](http://docs.google.com/javax/management/openmbean/OpenType.html#equals(java.lang.Object)) in class [OpenType](http://docs.google.com/javax/management/openmbean/OpenType.html)<[T](http://docs.google.com/javax/management/openmbean/ArrayType.html)> **Parameters:**obj - the object to be compared for equality with this ArrayType instance; if obj is null or is not an instance of the class ArrayType this method returns false. **Returns:**true if the specified object is equal to this ArrayType instance.**See Also:**[Object.hashCode()](http://docs.google.com/java/lang/Object.html#hashCode()), [Hashtable](http://docs.google.com/java/util/Hashtable.html)

### hashCode

public int **hashCode**()

Returns the hash code value for this ArrayType instance.

The hash code of an ArrayType instance is the sum of the hash codes of all the elements of information used in equals comparisons (i.e. dimension, elements' open type and primitive array flag). The hashcode for a primitive value is the hashcode of the corresponding boxed object (e.g. the hashcode for true is Boolean.TRUE.hashCode()). This ensures that t1.equals(t2) implies that t1.hashCode()==t2.hashCode() for any two ArrayType instances t1 and t2, as required by the general contract of the method [Object.hashCode()](http://docs.google.com/java/lang/Object.html#hashCode()).

As ArrayType instances are immutable, the hash code for this instance is calculated once, on the first call to hashCode, and then the same value is returned for subsequent calls.

**Specified by:**[hashCode](http://docs.google.com/javax/management/openmbean/OpenType.html#hashCode()) in class [OpenType](http://docs.google.com/javax/management/openmbean/OpenType.html)<[T](http://docs.google.com/javax/management/openmbean/ArrayType.html)> **Returns:**the hash code value for this ArrayType instance**See Also:**[Object.equals(java.lang.Object)](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [Hashtable](http://docs.google.com/java/util/Hashtable.html)

### toString

public [String](http://docs.google.com/java/lang/String.html) **toString**()

Returns a string representation of this ArrayType instance.

The string representation consists of the name of this class (i.e. javax.management.openmbean.ArrayType), the type name, the dimension, the elements' open type and the primitive array flag defined for this instance.

As ArrayType instances are immutable, the string representation for this instance is calculated once, on the first call to toString, and then the same value is returned for subsequent calls.

**Specified by:**[toString](http://docs.google.com/javax/management/openmbean/OpenType.html#toString()) in class [OpenType](http://docs.google.com/javax/management/openmbean/OpenType.html)<[T](http://docs.google.com/javax/management/openmbean/ArrayType.html)> **Returns:**a string representation of this ArrayType instance

### getArrayType

public static <E> [ArrayType](http://docs.google.com/javax/management/openmbean/ArrayType.html)<E[]> **getArrayType**([OpenType](http://docs.google.com/javax/management/openmbean/OpenType.html)<E> elementType)  
 throws [OpenDataException](http://docs.google.com/javax/management/openmbean/OpenDataException.html)

Create an ArrayType instance in a type-safe manner.

Multidimensional arrays can be built up by calling this method as many times as necessary.

Calling this method twice with the same parameters may return the same object or two equal but not identical objects.

As an example, the following piece of code:

ArrayType t1 = ArrayType.getArrayType(SimpleType.STRING);  
 ArrayType t2 = ArrayType.getArrayType(t1);  
 ArrayType t3 = ArrayType.getArrayType(t2);  
 System.out.println("array class name = " + t3.getClassName());  
 System.out.println("element class name = " + t3.getElementOpenType().getClassName());  
 System.out.println("array type name = " + t3.getTypeName());  
 System.out.println("array type description = " + t3.getDescription());

would produce the following output:

array class name = [[[Ljava.lang.String;  
 element class name = java.lang.String  
 array type name = [[[Ljava.lang.String;  
 array type description = 3-dimension array of java.lang.String

**Parameters:**elementType - the *open type* of element values contained in the arrays described by this ArrayType instance; must be an instance of either SimpleType, CompositeType, TabularType or another ArrayType with a SimpleType, CompositeType or TabularType as its elementType. **Throws:** [OpenDataException](http://docs.google.com/javax/management/openmbean/OpenDataException.html) - if elementType's className is not one of the allowed Java class names for open data.**Since:** 1.6

### getPrimitiveArrayType

public static <T> [ArrayType](http://docs.google.com/javax/management/openmbean/ArrayType.html)<T> **getPrimitiveArrayType**([Class](http://docs.google.com/java/lang/Class.html)<T> arrayClass)

Create an ArrayType instance in a type-safe manner.

Calling this method twice with the same parameters may return the same object or two equal but not identical objects.

As an example, the following piece of code:

ArrayType t = ArrayType.getPrimitiveArrayType(int[][][].class);  
 System.out.println("array class name = " + t.getClassName());  
 System.out.println("element class name = " + t.getElementOpenType().getClassName());  
 System.out.println("array type name = " + t.getTypeName());  
 System.out.println("array type description = " + t.getDescription());

would produce the following output:

array class name = [[[I  
 element class name = java.lang.Integer  
 array type name = [[[I  
 array type description = 3-dimension array of int

**Parameters:**arrayClass - a primitive array class such as int[].class, boolean[][].class, etc. The [getElementOpenType()](http://docs.google.com/javax/management/openmbean/ArrayType.html#getElementOpenType()) method of the returned ArrayType returns the [SimpleType](http://docs.google.com/javax/management/openmbean/SimpleType.html) corresponding to the wrapper type of the primitive type of the array. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if arrayClass is not a primitive array.**Since:** 1.6

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ArrayType.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PREV CLASS   [**NEXT CLASS**](http://docs.google.com/javax/management/openmbean/CompositeData.html) | [**FRAMES**](http://docs.google.com/index.html?javax/management/openmbean/ArrayType.html)    [**NO FRAMES**](http://docs.google.com/ArrayType.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#2et92p0) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: FIELD | [CONSTR](#2s8eyo1) | [METHOD](#26in1rg) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).