<meta-data>

SYNTAX:

```
<meta-data android:name="string"
           android:resource="resource specification"
           android:value="string" />
```

CONTAINED IN:

<activity>

<activity-alias>

<application>

<provider>

<receiver>

<service>

DESCRIPTION:

A name-value pair for an item of additional, arbitrary data that can be supplied to the parent component. A component element can contain any number of `<meta-data>` subelements. The values from all of them are collected in a single `Bundle` object and made available to the component as the `PackageItemInfo.metaData` field.

Ordinary values are specified through the `value` attribute. However, to assign a resource ID as the value, use the `resource` attribute instead. For example, the following code assigns whatever value is stored in the `@string/kangaroo` resource to the "zoo" name:

```
<meta-data android:name="zoo" android:value="@string/kangaroo" />
```

On the other hand, using the `resource` attribute would assign "zoo" the numeric ID of the resource, not the value stored in the resource:

```
<meta-data android:name="zoo" android:resource="@string/kangaroo" />
```

It is highly recommended that you avoid supplying related data as multiple separate `<meta-data>` entries. Instead, if you have complex data to associate with a component, store it as a resource and use the `resource` attribute to inform the component of its ID.

ATTRIBUTES:

android:name

A unique name for the item. To ensure that the name is unique, use a Java-style naming convention — for example, "`com.example.project.activity.fred`".

android:resource

A reference to a resource. The ID of the resource is the value assigned to the item. The ID can be retrieved from the meta-data Bundle by the `Bundle.getInt()` method.

android:value

The value assigned to the item. The data types that can be assigned as values and the Bundle methods that components use to

retrieve those values are listed in the following table:

Type	Bundle method
String value, using double backslashes (\\) to escape characters — such as "\\n" and "\\uxxxxx" for a Unicode character.	getString()
Integer value, such as "100"	getInt()
Boolean value, either "true" or "false"	getBoolean()
Color value, in the form "#rgb", "#argb", "#rrggbb", or "#aarrggbb"	getInt()
Float value, such as "1.23"	getFloat()

INTRODUCED IN:

API Level 1