# Mirror API

Mirror API是用来模型化程序的语义结构

**See:**   
[**描述**](#Description)

|  |  |
| --- | --- |
| **Packages** | |
| [**com.sun.mirror.apt**](mk:@MSITStore:E:\doc\手册\jdk6.ZH_cn.chm::/j2se6/jdk/api/apt/mirror/com/sun/mirror/apt/package-summary.html) | 用于在注释处理器和注释处理工具之间传递信息的类。 |
| [**com.sun.mirror.declaration**](mk:@MSITStore:E:\doc\手册\jdk6.ZH_cn.chm::/j2se6/jdk/api/apt/mirror/com/sun/mirror/declaration/package-summary.html) | 用于模型化程序元素声明的接口 |
| [**com.sun.mirror.type**](mk:@MSITStore:E:\doc\手册\jdk6.ZH_cn.chm::/j2se6/jdk/api/apt/mirror/com/sun/mirror/type/package-summary.html) | 用于模型化类型的接口 |
| [**com.sun.mirror.util**](mk:@MSITStore:E:\doc\手册\jdk6.ZH_cn.chm::/j2se6/jdk/api/apt/mirror/com/sun/mirror/util/package-summary.html) | 用于辅助处理声明和类型的实用工具 |

The Mirror API is used to model the semantic structure of a program. It provides representations of the entities declared in a program, such as classes, methods, and fields. Constructs below the method level, such as individual statements and expressions, are not represented.

Mirror是用来模型化程序的语义结构。他提供了在程序中声明的实体，像类、方法、字段的表示。但是像方法内部的局部变量、表达式是不能表示的

Also included is support for writing [annotation processors](mk:@MSITStore:E:\doc\手册\jdk6.ZH_cn.chm::/j2se6/jdk/api/apt/mirror/com/sun/mirror/apt/AnnotationProcessor.html) to examine and process the annotations of program elements. An annotation processor may, as an example, create new source files and XML documents to be used in conjunction with the original code.

也包括了实现注解处理器的支持，注解处理器是用来验证和处理程序元素的注解。一个注解处理器可能会做这样的事情来配合源码使用：创建一个源文件和xml文件。

**API特征**

A program is represented at the language level, rather than at the level of the virtual machine. Nested classes, for example, are handled as first-class constructs, rather than in the translated form understood by the VM. Both source code and compiled code (class files) may be modeled in this way.

这个是语言层次的，而不是虚拟机层次的。嵌套类，例如作为类的构造处理，而不是翻译虚拟机的理解。源代码和编译代码（class文件）都可以用这种方法模型化。

Programs are modeled in their static, or build-time, form. This differs from the [reflection](mk:@MSITStore:E:\doc\手册\jdk6.ZH_cn.chm::/j2se6/api/java/lang/reflect/package-summary.html?is-external=true) API, which provides run-time information about classes and objects.

The API does not provide direct support for generating new code.

**Declarations and Types**

The mirror API represents program constructs principally through the [Declaration](mk:@MSITStore:E:\doc\手册\jdk6.ZH_cn.chm::/j2se6/jdk/api/apt/mirror/com/sun/mirror/declaration/Declaration.html) interface and its hierarchy of subinterfaces in the package [com.sun.mirror.declaration](mk:@MSITStore:E:\doc\手册\jdk6.ZH_cn.chm::/j2se6/jdk/api/apt/mirror/com/sun/mirror/declaration/package-summary.html). A Declaration represents a program element such as a package, class, or method. The interface hierarchy is depicted [here](mk:@MSITStore:E:\doc\手册\jdk6.ZH_cn.chm::/j2se6/jdk/api/apt/mirror/com/sun/mirror/declaration/package-tree.html).

Types are represented by the [TypeMirror](mk:@MSITStore:E:\doc\手册\jdk6.ZH_cn.chm::/j2se6/jdk/api/apt/mirror/com/sun/mirror/type/TypeMirror.html) interface and its hierarchy of subinterfaces in the package [com.sun.mirror.type](mk:@MSITStore:E:\doc\手册\jdk6.ZH_cn.chm::/j2se6/jdk/api/apt/mirror/com/sun/mirror/type/package-summary.html). Types include primitive types, class and interface types, array types, type variables, and wildcards. The interface hierarchy is depicted [here](mk:@MSITStore:E:\doc\手册\jdk6.ZH_cn.chm::/j2se6/jdk/api/apt/mirror/com/sun/mirror/type/package-tree.html).

The API makes a clear distinction between declarations and types. This is most significant for generic types, where a single declaration can define an infinite family of types. For example, the declaration of java.util.Set defines the raw type java.util.Set, the parameterized type java.util.Set<String>, and much more. Only the declaration can be annotated, for example, and only a type can appear in a method signature.

A program being modeled may be incomplete, in that it may depend on an unknown class or interface type. This may be the result of a processing error such as a missing class file, or perhaps the missing type is to be created by an annotation processor. See [DeclaredType](mk:@MSITStore:E:\doc\手册\jdk6.ZH_cn.chm::/j2se6/jdk/api/apt/mirror/com/sun/mirror/type/DeclaredType.html) for information on how such unknown types are handled.

**Utilities and Tool Support**

The [com.sun.mirror.util](mk:@MSITStore:E:\doc\手册\jdk6.ZH_cn.chm::/j2se6/jdk/api/apt/mirror/com/sun/mirror/util/package-summary.html) package provides utilities to assist in the processing of declarations and types. Included is support for using the visitor design pattern when operating on declaration and type objects.

The [com.sun.mirror.apt](mk:@MSITStore:E:\doc\手册\jdk6.ZH_cn.chm::/j2se6/jdk/api/apt/mirror/com/sun/mirror/apt/package-summary.html) package supports the writing of annotation processors. It provides the mechanism for them to interact with an annotation processing tool.

**Since:**

1.5