Experimental Definitions

The `@experimental` annotation allows the definition of an API that is not guaranteed backward binary or source compatibility. This annotation can be placed on term or type definitions.

References to experimental definitions

Experimental definitions can only be referenced in an experimental scope. Experimental scopes are defined as follows:

1. The RHS of an experimental `def`, `val`, `var`, `given` or `type` is an experimental scope. Examples:

   ▶ Example 1
   ▶ Example 2
   ▶ Example 3
   ▶ Example 4
   ▶ Example 5
2. The signatures of an experimental `def`, `val`, `var`, `given` and `type`, or constructors of `class` and `trait` are experimental scopes. Examples:

   ▶ Example 1
3. The `extends` clause of an experimental `class`, `trait` or `object` is an experimental scope. Examples:

   ▶ Example 1
4. The body of an experimental `class`, `trait` or `object` is an experimental scope. Examples:

   ▶ Example 1
5. Annotations of an experimental definition are in experimental scopes. Examples:

▶ Example 1

6. Any code compiled using a *Nightly* or *Snapshot* version of the compiler is considered to be in an experimental scope. Can use the `-Yno-experimental` compiler flag to disable it and run as a proper release.

In any other situation, a reference to an experimental definition will cause a compilation error.

Experimental inheritance

All subclasses of an experimental `class` or `trait` must be marked as `@experimental` even if they are in an experimental scope. Anonymous classes and SAMs of experimental classes are considered experimental.

We require explicit annotations to make sure we do not have completion or cycles issues with nested classes. This restriction could be relaxed in the future.

Experimental overriding

For an overriding member `M` and overridden member `O`, if `O` is non-experimental then `M` must be non-experimental.

This makes sure that we cannot have accidental binary incompatibilities such as the following change.

```
class A:
  def f: Any = 1
class B extends A:
-  @experimental def f: Int = 2
```

Test frameworks

Tests can be defined as experimental. Tests frameworks can execute tests using reflection even if they are in an experimental class, object or method. Examples:

▶ Example 1

‹ TypeTest Other ... ›