

Java SE Documentation

Static Import



In order to access static members, it is necessary to qualify references with the class they came from. For example, one must say:

```
double r = Math.cos(Math.PI * theta);
```

In order to get around this, people sometimes put static members into an interface and inherit from that interface. This is a bad idea. In fact, it's such a bad idea that there's a name for it: the *Constant Interface Antipattern* (see *Effective Java* Item 17). The problem is that a class's use of the static members of another class is a mere implementation detail. When a class implements an interface, it becomes part of the class's public API. Implementation details should not leak into public APIs.

The static import construct allows unqualified access to static members *without* inheriting from the type containing the static members. Instead, the program *imports* the members, either individually:

```
import static java.lang.Math.PI;
```

or en masse:

```
import static java.lang.Math.*;
```

Once the static members have been imported, they may be used without qualification:

```
double r = cos(PI * theta);
```

The static import declaration is analogous to the normal import declaration. Where the normal import declaration imports classes from packages, allowing them to be used without package qualification, the static import declaration imports static members from classes, allowing them to be used without class qualification.

So when should you use static import? **Very sparingly!** Only use it when you'd otherwise be tempted to declare local copies of constants, or to abuse inheritance (the Constant Interface Antipattern). In other words, use it when you require frequent access to static members from one or two classes. If you overuse the static import feature, it can make your program unreadable and unmaintainable, polluting its namespace with all the static members you import. Readers of your code (including you, a few months after you wrote it) will not know which class a static member comes from. Importing *all* of the static members from a class can be particularly harmful to readability; if you need only one or two members, import them individually. Used appropriately, static import can make your program *more* readable, by removing the boilerplate of repetition of class names.