

Java Mapping for Sequences

A Slice [sequence](#) maps to a Java array. This means that the Slice-to-Java compiler does not generate a separate named type for a Slice sequence.

For example:

Slice

```
sequence<Fruit> FruitPlatter;
```

This definition simply corresponds to the Java type `Fruit[]`. Naturally, because Slice sequences are mapped to Java arrays, you can take advantage of all the array functionality provided by Java, such as initialization, assignment, cloning, and the `length` member. For example:

Java

```
Fruit[] platter = { Fruit.Apple, Fruit.Pear };  
assert(platter.length == 2);
```

[Alternate mappings](#) for sequence types are also possible.

See Also

- [Sequences](#)
- [Java Mapping for Enumerations](#)
- [Java Mapping for Structures](#)
- [Java Mapping for Dictionaries](#)
- [Customizing the Java Mapping](#)