

# NewTypeSupport

## New Type Support

This activity creates support for new types introduced into the Java language. For JDK 1.4, three types that appear to be of interest were introduced: Currency, [LinkedHashMap](#), and [LinkedHashSet](#).

## General Issues

- New test cases need to be written for these classes. This is more work than it appears, because we need to look at the existing RI tests and see which of these should be converted to TCK tests. There are a number of tests that verify that adding and removing elements from collections and adding and removing entries from maps result in the correct behavior. These were never converted to TCK tests. So now is the time to take a careful look and make them TCK tests.
- The classes are only available in JDK 1.4 and above. So we need a different strategy for loading the support. Currently, the constructor of the [ObjectTranscriber](#) constructs the association map between classes and the corresponding transcriber. The constructor must work regardless of whether the classes are available in the environment. Probably we should define a new class for the JDK 1.4 transcribers ([ObjectTranscriber14Initializer](#)) and another new class for JDK 1.5 transcribers ([ObjectTranscriber15Initializer](#)). These classes cannot contain static references to the new classes; otherwise an [ExceptionInInitializer](#) will occur. It might be best to define yet another class that can be called only if running in the appropriate environments. The [ObjectTranscriber](#) can call these new classes, and the new classes can determine if the environment is suitable. If not, they silently return. If they do have access to the classes, then they can construct the transcriber instance and call back into the [ObjectTranscriber](#) with the new class/transcriber map entry. This interface needs to be designed. Perhaps a new method on [ObjectTranscriber](#) needs to be defined, but be careful because the [ObjectTranscriber](#) is in its constructor. It might work to have the transcriber map be a static member that is initialized to null in the initializer of [ObjectTranscriber](#), is partially populated by the [ObjectTranscriber](#) constructor, and is finished by the 14 and 15 initializers.

## Currency

The strategy for Currency is to store the currency code persistently as a String. This can be done in a [CurrencyTranscriber](#) in which the output sent to the stream is the result of calling `getCurrencyCode` on the instance. Similarly, when reading an instance from the stream, the static method `Currency.newInstance(String)` that takes the currency code can be called.

## LinkedHashMap

The strategy for [LinkedHashMap](#) is similar to [HashMap](#).

## LinkedHashSet

The strategy for [LinkedHashSet](#) is similar to [HashSet](#).