

SerialVersionUID values, since the default SerialVersionUID computatⁿ is highly sensitive to class detail that may vary depending on compiler implementⁿ, many details including fields & in your class, their access modifiers, the interface they implement and even different implementⁿ, any change on class or using different compiler may result in different SerialVersionUID, which may eventually step reloading serialized data.

It's too risky to rely on Java serialization Mechanism for generating this id, and that's why it's recommended to declare explicit SerialVersionUID in your Serializable class and can thus result in unexpected InvalidClassExceptⁿ during deserializⁿ. Therefore, to guarantee a consistent SerialVersionUID value across different Java compiler implementⁿ, a serializable class must declareⁿ explicit SerialVersionUID value.

• You can use JDK's 'serialver' tool to generate SerialVersionUID for classes. This is particularly useful for evolving classes, it returns SerialVersionUID for classes in format easy to copy. You can use 'serialver' tool

SDK tool as shown in below example:-

```
} $ serialver -classpath . emp  
{ Emp: static final long serialVersionUID=----
```

Now, we know what is `SerialVersionUID` and why it's important to declare it in Serializable class, it's time to revise some of the important facts related to Java `SerialVersionUID`.

1) `SerialVersionUID` is used to version serialized data. You can only de-serialize a class if its `SerialVersionUID` matches with the serialized instance.

2) When we don't declare `SerialVersionUID` in our class, Java runtime generates it for us, but that process is sensitive to many class meta data including no. of fields, type of fields, access modifiers of fields, interface implemented by class etc.

3) It is recommended to declare `SerialVersionUID` as private static long variable to avoid default mechanism.

4) You can even use 'serialver' tool from SDK to generate `SerialVersionUID` for classes in Java. It also has

has a GUT, which can be
enable by passing -show
parameter.

5) I43 Serialization best prac-
- tice in Java to explicitly
declare serialVersionUID, to
avoid any issues during de-
- serialization especially if you
are running a client server
application which relies on
serialized data eg - RMI.