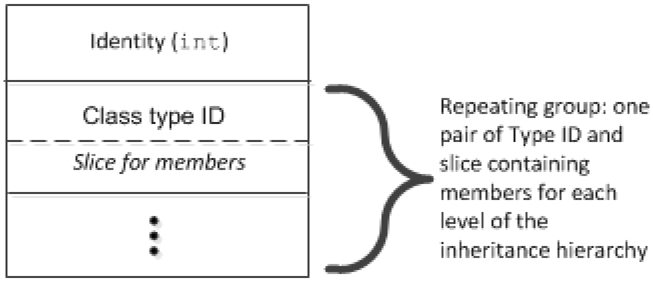
# Data Encoding for Classes

The marshaling for classes is complex, due to the need to deal with the pointer semantics for graphs of classes, as well as the need for the receiver to slice classes of unknown derived type. In addition, the marshaling for classes uses a type ID compression scheme to avoid repeatedly marshaling the same type IDs for large graphs of class instances.

Classes are marshaled similar to exceptions: each instance is divided into a number of pairs containing a type ID and a slice (one pair for each level of the inheritance hierarchy) and marshaled in derived-to-base order. Only data members are marshaled — no information is sent that would relate to operations. Unlike exceptions, no header byte precedes a class. Instead, each marshaled class instance is preceded by a (non-zero) positive integer that provides an identity for the instance. The sender assigns this identity during marshaling such that each marshaled instance has a different identity. The receiver uses that identity to correctly reconstruct graphs of classes. The overall marshaling format for classes is shown below:


Identity (int)
Class type ID
Slice for members
Repeating group: one pair of Type ID and slice containing members for each level of the inheritance hierarchy

*Marshaling format for classes.*

## Topics

- Data Encoding for Class Type IDs
- Simple Example of Class Encoding
- Data Encoding for Class Graphs

### See Also

- Classes
- Type IDs
- Data Encoding for Exceptions
- Basic Data Encoding