

# ResourcesUserGuideMessages

[Home](#) [Wiki](#) [Guide](#) [Getting Started](#) [Messages](#) [Standard](#) [Creating](#) [API](#) [Source](#)

## 2. Using Message and MessageList

### 2.1 Message

**Message** is an encapsulation of an individual message stored in a **MessageList** object, consisting of a message key (to be used to look up message text in an appropriate Resources implementation) plus up to four placeholder objects that can be used for parametric replacement in the message text.

Message defines two methods:

- **getKey()** - return the message key
- **getValues()** - return an array of replacement values

**BasicMessage** is a basic implementation of a **Message** and provides convenience constructors for creating a Message with 1, 2, 3 or 4 replacement values.

For example, to create a message with two replacement values (100 and 200 in this case):

```
Message message = new BasicMessage("error.range", "100", "200");
```

### 2.2 Message List

**MessageList** a class that encapsulates messages. Messages can be either global or they are specific to a particular bean property. Each individual message is described by a **Message** object. **MessageList**, as its name implies, has similarities to a *java.util.List* and provides methods for adding and retrieving messages.

**BasicMessageList** is a basic implementation of a **MessageList**.

```
MessageList list = new BasicMessageList();
```

Messages can be stored under a **Global Message Key** or they can be stored under a specific **Property Key**

```
list.add(message);           // stored for the Global Message Key
list.add("OrderNo", message); // stored for property "OrderNo"
```

Sets of messages can be retrieved from the list either for a property, for the global message key or all messages:

```
Iterator allMsgs      = list.get();
Iterator orderNoMsgs = list.get("OrderNo");
Iterator globalMsgs  = list.get(getGlobalMessageKey());
```

### 2.3 Messages

**Messages** is a *wrapper* around any **Resources** object that performs message string lookups from the Resources instance, and parameter replacement via **j ava.text.MessageFormat**. For convenience, the same functionality is also available via static methods that accept a Resources parameter.

First you need to create the **Messages** object...

```
Resources resources = factory.getResources("Bar", "file:c:/myapp/foo/Bar");
Messages messages = new Messages(resources);
```

For example to retrieve the text for a message key "error.maximum" for the US locale with one replacement argument...

```
String text = messages.get(Locale.US, "error.maximum", 1000);
```

Alternatively you can use the static methods...

```
String text = Messages.get(resources, Locale.US, "error.maximum", 1000);
```