

Chapter 3 Bean Events

1. If an event class is named *XEvent*, its listener interface should be named *XListener*. Consequently, the registration methods in the source component should be named `addXListner(XListener l)` and `removeXListener(XListener l)`.
2. Yes. Yes.
3. Event *e* is generated in the constructor using
`e = new TickEvent(this);`
It is fired by invoking the `process(e)` method.
4. A component inspector is displayed with the UI designer.
5. The registration method for unicasting throws `TooManyListenersException`. The registration method for multicasting uses a vector to store all listeners.
6. See Page 96.
7. A convenience adapter implements the appropriate listener interface. It is convenient, because all the handlers are provided with an empty body. You can use these convenient adapters without implementing the handlers you are not interested.
8. Use the constructor:
`new ActionEvent(this,`
`ActionEvent.ACTION_PERFORMED, null));`
9. Forte generates the inner class and registers it with the button object.
10. It is special for inner class. An anonymous inner class always uses the default constructor from its superclass to create an instance. If an anonymous inner class implements an interface, the default constructor is `Object()`.