Base Exception 1.0 Component Specification

1. Design

This component defines three exception and error classes, which extend Exception, RuntimeException and Error. They are intended to serve as the base exception and error classes for all exceptions and errors defined by an application. Used this way, the Base Exception component can provide additional exception-related functionality to an entire application.

In particular this component provides "chained exception" functionality like that found in JDK 1.4 -- Throwables can be attached to this component's exception and error classes. This functionality works in JDK 1.2 and later, and also integrates with JDK 1.4's implementation of chained exceptions.

The API is designed to resemble JDK 1.4's Throwable API as much as possible, for forwards compatibility. That means that this component's exception and error classes can print an entire chain of stack traces, for example.

1.1 Design Patterns

Delegate:
CauseUtils delegates to one of two implementations of CauseUtilsDelegate depending on the current JVM version.

1.2 Industry Standards

Mirrors the JDK 1.4 exception/error class API and behavior

1.3 Required Algorithms

Printing a chained stack trace in BaseException/RuntimeException/Error printStackTrace() methods doesn't need to be any more complex than this:

```
super.printStackTrace()
if(cause is not null)
    print "Caused by:"
          cause.printStackTrace()
```

Likewise for creating the exception's message in getMessage().

This is an implementation note more than an algorithm, but it is important enough to bear repeating:
CauseUtils must not reference CauseUtilsDelegateImpl14 directly (or else the component will not load under JDK 1.3 or earlier). It must load it by reflection, using Class.forName().newInstance().

1.4 Component Class Overview

BaseException:
Subclass of Exception; base class for checked exceptions.

BaseRuntimeException:
Subclass of RuntimeException; base class for unchecked exceptions.

BaseError:
Subclass of Error; base class for errors.

CauseUtils:
Utility class with methods that retrieve the cause of an exception, regardless of its nature (BaseException, JDK 1.4 Throwable)

CauseUtilsDelegate, CauseUtilsDelegateImpl13, CauseUtilsDelegateImpl14:
Package-private interface and two implementations that get the cause of a Throwable in a way that functions under JDKs prior to 1.4, and 1.4 and later, respectively.

1.5 Component Exception Definitions

NullPointerException:
Thrown if printStackTrace() is called with a null argument.

IllegalArgumentException:
Thrown when initCause() on an instance with a reference to the same instance (A Throwable can't be its own cause).

IllegalStateException:
Thrown when initCause() is called after a cause has already been set.

2. Environment Requirements

2.1 TopCoder Software Components:

- None

2.2 Third Party Components:

- None

3. Usage Notes

3.1 Required steps to test the component

- Extract the component distribution.
- Execute ‘ant test’ within the directory that the distribution was extracted to.

3.2 Required steps to use the component

An application must first extend one of the three exception/error classes from this component, as appropriate. It is recommended that an application provide the four standard constructors, like so:

```
public class MyBusinessException extends BaseException {
        public MyBusinessException() {
                super();
        }
        public MyBusinessException(String message) {
                super(message);
        }
        public MyBusinessException(String message, Throwable cause) {
                super(message, cause);
        }
        public MyBusinessException(Throwable cause) {
                super(cause);
        }
}
```

It can then be thrown like any other exception, along with an optional "cause":

```
...
} catch (SQLException sqle) {
        throw MyBusinessException("Something's wrong", sqle);
}
...
```

And then, for example, the entire chained stack trace can be printed:

```
...
} catch (MyBusinessException mbe) {
        mbe.printStackTrace();
}
```

3.3 Demo

None