

## ALL – IN – ONE FAQ

### Language:

1) What are the differences between the == operator and the equals() method?

Well, first off, == is a fundamental operator in the language. The result type of the expression is a boolean. For comparing boolean types, it compares the operands for the same truth value. For comparing reference types, it compares the operands for the same reference value (i.e., refer to the same object or are both null). For numeric types, it compares the operands for the same integer value or equivalent floating point values.

In contrast, equals() is an instance method which is fundamentally defined by the java.lang.Object class. This method, by convention, indicates whether the receiver object is "equal to" the passed in object. The base implementation of this method in the Object class checks for reference equality. Other classes, including those you write, may override this method to perform more specialized equivalence testing.

2) How can I force garbage collection to take place?

Strictly, you can't force it.

You can, however, call System.gc(), which is a "hint" to the runtime engine that now might be a good time to run the GC. Some implementations take that hint to heart and some don't.

3) When should I use an interface instead of an abstract class?

There are two primary axes of "inheritance" in object-oriented languages like Java. "Implementation" inheritance is where the sub-class inherits the actual code implementation from the parent. "Interface" inheritance is where the "sub-class" adheres to the public interface of the "parent".

Alas, Java actually mixes the two notions together a bit... Java interfaces are nice and clean -- when you "implement" an interface, you are stipulating that your class adheres to the "contract" of the interface that you specified. Java class inheritance isn't so clean -- when you sub-class in Java you are getting both the code inheritance but you are also stipulating that your sub-class adheres to the "contract" of the interface of the parent class.

Abstract classes in Java are just like regular Java classes but with the added constraint that you cannot instantiate them directly. In terms of that added constraint, they are basically classes which don't actually implement all of the code specified by their "contract".

So, it's generally considered good OO practice to specify the "contract" which you want to adhere to via Java interfaces. Then use normal Java class inheritance primarily for code reuse purposes. Use abstract Java classes when you want to provide some standard base code but want/need to force the user's of your class to complete the implementation (i.e., you create a skeleton implementation and the sub-classes must flesh it out).

#### 4) What are the differences between instance and class variables?

Instance variables have separate values for each instance of a class. Class variables maintain a single shared value for all instances of the class, even if no instance object of that class exists.

You would use the static keyword to change an instance variable into a class variable.

Both instance and class variables are declared at the class level, not within methods:

```
public class Foo {  
    static private int count; // class variable  
    String name; // instance variable  
    private void Bar() {  
        int halfCount; // local variable  
    }  
}
```

Note also that classes are loaded by classloaders therefore class variables are unique on a per-classloader basis.

#### 5) What is an overloaded method?

Overloaded methods are multiple methods in the same class that share the same name but have different parameter lists. Overloaded methods *cannot* have the same parameter lists with different return types.

#### 6) What is a final class?

When you declare a class to be final, it can never be subclassed. This is usually done for security or performance reasons.

```
public final class Math {  
    // ...  
}
```

7) What is a final method?

When you declare a method to be final, it can never be overridden in subclasses. This is usually done for security or performance reasons, providing additional information to the JIT compiler or HotSpot. One word of caution -- making methods final restricts later design decisions.

8) Why do I not have to import the classes in the java.lang package?

The classes in the java.lang package are so important that the compiler implicitly imports all the classes there. This ensures classes in the unnamed package are not accidentally used instead of the key classes and interfaces.

9) Can I override a method that doesn't throw exceptions with one that does?

Overriding methods cannot change the signature of the overridden method. All exceptions are included in the signature of the method except for RuntimeException and its subclasses. So, the only exceptions you may add to an overriding method must be a (sub-class of) RuntimeException.

10) Can I restart a stopped thread?

In short, no. Once a thread is stopped, it cannot be restarted. Keep in mind though that the use of the stop() method of Thread is deprecated and should be avoided.

11) How do I have one thread wait for another thread to finish before continuing?

You can wait for a thread to finish by calling its join() method. For instance, in the following code, the current thread will wait until thread2 finishes before printing Done.

```
thread2.start();  
// do more stuff here  
thread2.join();  
System.out.println("Done");
```

12) What is a method's signature?

The signature of a method is the combination of the method's name along with the number and types of the parameters (and their order).

13) Why does "float f = 3.5;" generate a compile-time error?

The constant value 3.5 is recognized by the compiler to be a double rather than a float. You must place an f after the 3.5 for it to be recognized as a floating point constant value:

```
float f = 3.5f;
```

14) What is the difference between the & and && operators?

It depends on the type of the arguments...

For integer arguments, the single ampersand("&") is the "bit-wise AND" operator. The double ampersand("&&") is not defined for anything but two boolean arguments.

For boolean arguments, the single ampersand constitutes the (unconditional) "logical AND" operator while the double ampersand("&&") is the "conditional logical AND" operator. That is to say that the single ampersand always evaluates both arguments whereas the double ampersand will only evaluate the second argument if the first argument is true.

For all other argument types and combinations, a compile-time error should occur.

15) How can I cast an Object reference to an array reference?

Pretty much just like any other cast:

```
MyArrayRef = (MyArrayType []) objectRef;
```

16) How is Java's "byte code" different from other codes (like source code, native code, etc.)?

Java's byte code is what the typical Java compilers create in the .class files. It is a binary language that is defined for the Java Virtual Machine (JVM). The JVM is the abstract machine which runs the byte codes (analogously to how an Intel 80386 runs .obj files).

17) Can an anonymous class have static members?

With one exception, anonymous classes cannot contain static fields, methods, or classes. The only static things they can contain are static final constants.

18) When I create anonymous classes, what access modifier do they get?

Anonymous classes get the default, unnamed access modifier. You cannot make them private, protected, public, or static.

19) How do I create a constructor for an anonymous class?

Since anonymous classes don't have names, you cannot create a constructor by just using the name of the class. What you can do instead is to use an "instance initializer". Instance initializers basically look like methods with no names or return values:

```
ActionListener listener = new ActionListener {  
    {  
        System.out.println("Instance Initializer");  
    }  
    public void actionPerformed(ActionEvent event) {  
        ...  
    }  
};
```

20) When should I use a "private" constructor?

The most common use that I see for private (and protected) constructors is in implementing singletons and some types of object factories. Basically, you use a static method to manage the creation of one (or more) instances of the class.

21) I'm trying to use an AWT List and the List collection in my program. How do I overcome the class name conflict?

In such a case of name conflict:

If you need only one definition but your type-import-on-demand (using import xxx.yyy.\*;) causes the conflict, use explicit import for the reference you need.

```
import java.awt.*;  
import java.util.List; // if you need this List in your program.
```

If you need both in the same Java file, you will have to explicitly refer at least one.

```
import java.awt.*;
import java.util.List; // you can use "List" for this one.
....
    java.awt.List myAwtList = new java.awt.List();

    List myUtilList = new List();
```

For clarity, I would suggest to explicitly manage both in that case.

## 22) How can I do a deep clone of an object?

The default/conventional behavior of `clone()` is to do a shallow copy. You can either override `clone()` to do a deep copy or provide a separate method to do a deep clone.

The simplest approach to deep cloning is to use Java serialization, where you serialize and deserialize the object and return the deserialized version. This will be a deep copy/clone, assuming everything in the tree is serializable. If everything is not serializable, you'll have to implement the deep cloning behavior yourself.

Assuming everything is serializable, the following should create a complete deep copy of the current class instance:

```
ByteArrayOutputStream baos = new ByteArrayOutputStream();
ObjectOutputStream oos = new ObjectOutputStream(baos);
oos.writeObject(this);
ByteArrayInputStream bais = new ByteArrayInputStream(baos.toByteArray());
ObjectInputStream ois = new ObjectInputStream(bais);
Object deepCopy = ois.readObject();
```

## 23) What are the differences between the built-in, primitive types (like `int`) and the equivalent, non-primitive object-wrapper classes (like `Integer`)?

The elemental types (`int`, `long` etc.) are fundamental parts of the Java language - they're reserved words. They directly represent a quantity, a character, etc.

The `java.lang` classes (e.g. `Void`, `Integer`, `Long` etc.) serve two functions - first, they provide utility methods (generally static to those classes) for doing commonplace tasks like turning an integer into a `String` and vice versa.

The second use for the `java.lang` classes is as a wrapper for the elemental types - this is needed because Java represents all collections (other than arrays) in terms of `java.lang.Object` - so e.g., a `Vector` can only contain objects, not elemental types. If you wanted a `Vector` or `Hashtable` of ints, you'd have to wrap

each int in a `java.lang.Integer`, and then unwrap the int when you retrieved values from the collection.

Equally, if you wanted to define a method that returned either a `String` or an `int`, you would define the function as returning a `java.lang.Object`; and then in practice you'd return either a `java.lang.String` or a `java.lang.Integer`, as appropriate.

24) In the main method of a class (or any static method), how can I find out what class I am in?

To get the `Class` object associated with that class, you have to reference the name of the class directly and use the `.class` keyword. I.e., if your class is named `Foo` use `Foo.class` to get the `Class` object for `Foo`.

25) How do I get a list of the files in a `.zip` file from Java?

You can get a `java.util.Enumeration` of `java.util.zip.ZipEntry` objects by using the `java.util.zip.ZipFile.entries()` method.

26) What is a JIT compiler?

A *Just-In-Time* compiler is one way to implement the Java Virtual Machine. Rather than the typical, pure interpreter, a JIT compiler will, on demand, compile the platform independent Java byte codes into platform specific executable code, in real-time. The first time through, the byte codes are compiled before being executed. Future passes will use a cached version of the compiled code. This tends to result in a performance increase of 10-100 times, depending upon the task at hand, over purely interpreted byte codes.

27) What's a `.jar` file?

A JAR file or Java ARchive is a way of bundling multiple files together as a single file. Like a ZIP file, they support compression and retain the directory structure of the files. They can be used for delivering things like JavaBeans and Applets, and usually include support files like image and sound files, in addition to the actual Java classes. In addition, they can be digitally signed so a user can grant privileges to allow the software to perform operations not normally permitted from untrusted software.

To create a JAR file, use the `jar` tool that comes with the JDK.

28) What are the differences between casting to `String`, using the `toString()` method, and using the `String.valueOf()` method?

Casting only transforms the reference to the object from one type to another. Casting does nothing to the underlying object itself. The compiler and/or the runtime engine will verify the fact that the reference that you're casting to a reference to a String is, in fact, referring to an object which is (type compatible with) a String.

The `String.valueOf()` method allows for the creation of a String representation of the built-in, primitive data types such as int, boolean, float, etc.

The `.toString()` method is a method which all Java classes inherit from the `java.lang.Object` root class (and which many classes override). The `.toString()` method of each class is supposed to create an appropriate String representation of objects of that class.

29) What is the difference between a Java compiler and a Java interpreter?

Typically, when used in that generic manner, the term *Java compiler* refers to a program which translates Java language source code into the Java Virtual Machine (JVM) *bytecodes*. The term *Java interpreter* refers to a program which implements the JVM specification and actually executes the bytecodes (and thereby running your program).

30) Why are the methods `wait()` and `notify()` defined in the `Object` class when they are used with `Threads`?

The `wait()` and `notify()` methods are object-specific. The `wait()` method suspends the current thread of execution, and tells the object to keep track of the suspended thread. The `notify()` method tells the object to wake up the suspended threads that it is currently keeping track of. Since `wait()` and `notify()` are object specific, they must be used within code that is synchronized on the object in question.

It is also important to use state variables when using `wait()` and `notify()`, as threads can be woken up by conditions other than `notify()`.

`suspend()` is similar to `wait()` but does not add the thread to an object's wait list.

31) How to run a batch file/native application from Java?

Use `Runtime.exec(String pathToBatchFile)`.

32) What are the differences between "JVM" and "JRE"?



Technically, the term "JVM" specifies the *Java Virtual Machine*. I.e., the "JVM" is the definition of the abstract, stack-based computational engine used to execute Java bytecodes. The term "JRE" stands for the *Java Runtime Environment* (if you follow Sun's nomenclature) or, interchangeably, *Java Runtime Engine*. I.e., a JRE is a specific implementation of the the JVM, including the core libraries.

Alas, in practice, many people use "JVM" for both cases. So, be careful to understand the context that people are talking about when using the term "JVM" to distinguish between the abstract definition and a specific implementation.

33) What's the difference between "JDK" and "JRE"?

The "JDK" is the *Java Development Kit*. I.e., the JDK is bundle of software that you can use to develop Java based software. The "JRE" is the *Java Runtime Environment*. I.e., the JRE is an implementation of the Java Virtual Machine which actually executes Java programs.

Typically, each JDK contains one (or more) JRE's along with the various development tools like the Java source compilers, bundling and deployment tools, debuggers, development libraries, etc.

34) What are the differences between *checked* and *unchecked* exceptions?

A checked exception is any subclass of Exception (or Exception itself), excluding class RuntimeException and its subclasses.

Making an exception checked forces client programmers to deal with the possibility that the exception will be thrown. eg, IOException thrown by java.io.FileInputStream's read() method

Unchecked exceptions are RuntimeException and any of its subclasses. Class Error and its subclasses also are unchecked.

With an unchecked exception, however, the compiler doesn't force client programmers either to catch the exception or declare it in a throws clause. In fact, client programmers may not even know that the exception could be thrown. eg, StringIndexOutOfBoundsException thrown by String's charAt() method.

Checked exceptions must be caught at compile time. Runtime exceptions do not need to be. Errors often cannot be, as they tend to be unrecoverable.

35) What are *mutable* and *immutable* objects in Java?

As per the dictionary, *mutable* objects are objects which can change their values and *immutable* objects are objects which cannot have their values changed.

36) Is it possible to start an application (from command line) from a method other than main?

No, it is not possible to start an application from a method other than main, as the runtime system will look for a method i.e., public static void main. If the method is not found we get an error by name java.lang.NoSuchMethodError.

37) Why can't I declare a static method in an interface?

Because all methods declared in an *interface* are (implicitly) also "abstract" methods since, by definition, they do *not* define the (implementation of) the method.

38) What's the difference between JavaScript and Java?

JavaScript is an object-oriented scripting language that allows you to create dynamic HTML pages, allowing you to process/validate input data and maintain data, usually within the browser. Originally created by Netscape for use with their Navigator browser, Microsoft reverse engineered the technology, added their own variants and defined JScript.

On the other hand, "Java" is a programming language, core set of libraries, and virtual machine platform that allows you to create compiled programs that run on nearly every platform, without distribution of source code in its raw form or recompilation. While the two share part of their names in common, they are really two completely different programming languages/models/platforms. Yes, they can communicate with each other through technologies like LiveScript. However, you should really consider them as two completely separate technologies.

39) How can I determine if a file is a Java class file?

If the first four bytes of a file are *not* 0xCAFEBAE then you do *not* have a Java class file. This testing is done as part of the first pass of the Java verifier.

40) What are the differences between PATH and CLASSPATH environment variables?

The PATH environment variable is typically something that the operating system uses to find executable files. See the system manuals for your platform for more information.

The CLASSPATH environment variable is typically something that implementations of the Java Virtual Machine (JVM) use to find Java class files.

41) Can a Java class be *static*?

Yes and no. The static modifier is only applicable to so-called *member classes* -- i.e., classes which are directly enclosed within another class.

42) What does it mean to override a method?

If a subclass shares a method name with identical arguments and return type, it overrides the method of its superclass. Overridden methods may differ in exceptions thrown by throwing a more specific exception type or none at all.

43) How do I find out how much system memory is available in my system?

The Runtime class has two methods to help you determine memory resources: `freeMemory()` and `totalMemory()`. The `freeMemory()` method reports the free bytes in Java's memory space. The `totalMemory()` reports the total bytes in Java's memory space.

44) What is the difference between the stream tokenizer and string tokenizer?

You can think of the `StringTokenizer` as a specialized form of `StreamTokenizer`, where each token that is returned is a `String`. Depending upon what you want to do with the resultant tokens and/or how complex of a parse job you want to deal with lets you decide which to use.

The `StringTokenizer` parses a string into a series of tokens. It is then your responsibility to figure out what each token returned is, beyond the definition of a token separated by delimiters. Usually, you just treat things as words when using `StringTokenizer`.

On the other hand, a `StreamTokenizer` allows you to ask is the next token a number, word, quoted string, end-of-file, end-of-line, comment, or whitespace. In this case, the `StreamTokenizer` is smarter, though can be considerably harder to use.

45) Can I override the `equals()` method to take a parameter other than `Object`?

If you are trying to override a method with a different argument list, that isn't overriding, but overloading. In order for the method to be properly overridden, and thus called appropriately, the subclass method signature must explicitly match that of the superclass. In the case of `equals()` that means the return type must be `boolean` and the argument must be `Object`.

46) Should I declare the constants in my interface as public final static, and my methods as public abstract?

The use of these modifiers in interfaces is completely optional. The following point out the relevant sections of the Java Language Specification:

- According to the section 9.1.5: *All interface members are implicitly public.*
- According to section 9.3: *Every field declaration in the body of an interface is implicitly public, static, and final*
- According to section 9.4: *Every method declaration in the body of an interface is implicitly public and Every method declaration in the body of an interface is implicitly abstract*

47) Why doesn't the Java runtime complain if the main() method isn't declared public?

With Sun's reference implementation, they introduced this bug into the 1.2 implementation and haven't fixed it through the 1.3 release. The main() method is supposed to be public static void but it seems any access modifier works fine.

48) Why are there so many different programming languages?

As is the case with most things in life, I believe there are multiple reasons for why we have so many programming languages. Here are what I believe to be the more significant reasons:

- Programming languages vary with respect to the speed with which programs written in them can be executed. So if you have a real-time application, you'd choose a language that would be capable of delivering results under the applicable time constraints. So for a problem in dynamic control, you might choose, say, C over C++ or Java.
- Certain application domains require programming languages that are specifically targeted for those applications. Cobol, for example, represents a language that was developed specifically for business applications. It is easy to learn by people without advanced degrees in computer science and it is efficient for what it was designed to do.
- Much early programming dealt with solving numerically intensive problems, such as problems encountered in scientific calculations. Fortran emerged as a favorite of many for such applications. I believe it continues to be a widely used language for solving numerically intensive problems on supercomputers.
- Many programming languages that have emerged from academic laboratories are a result of researchers trying to mimic certain aspects of human cognition. Languages like Lisp and Prolog fall in this category.
- Another reason for why we have so many programming languages is purely evolutionary. Consider, for example, the evolution from C to C++ and then on to

Java. As it began to be realized that we needed richer representations for our concepts and as we sought ways to make large programs more easily extensible and maintainable, the concepts of object-oriented programming came into existence. Then as we better understood what was meant by object-oriented programming through pioneering languages like Smalltalk, C led to the development of C++. And then as it dawned on us that the wide generality of C++ (with regard to inheritance, operator overloading, and other issues) could become a liability in some cases, Java emerged.

- As the role of computers as "information presentation devices" proliferated in the society, there emerged a concomitant need for languages designed specifically for formatting the visual display of information. Languages such as HTML, XML, and their variants are fulfilling those needs.

I suppose what it all means is that as we continue to deploy computers for solving previously unaddressed problems, we will try to use the languages we already know. But should they fall short of our needs for whatever reason, we as a society of programmers will invent new languages.

49) How can I prohibit methods from being overloaded?

The only relatively surefire way that I know of is to make the entire class final. I.e., since there's no way to subclass that class then nobody can overload that method.

50) How can I determine the version of the JRE at runtime?

Check the system properties:  
`System.getProperties().getProperty("java.version");`

51) What is a "marker" interface?

A so-called marker interface is a Java interface which doesn't actually define any fields. It is just used to "mark" Java classes which support a certain capability -- the class marks itself as implementing the interface. For example, the [java.lang.Cloneable](#) interface.

52) What is a static block and how should I use it?

Basically, static blocks are blocks defined within the body of a class using the static keyword but which are not inside any other blocks. I.e.,

```
public class shme
{
    static int foo = 12345;
```

```

    static
    {
        foo = 998877;
    }
}

```

The primary use of static initializers blocks are to do various bits of initialization that may not be appropriate inside a constructor such that taken together the constructor and initializers put the newly created object into a state that is completely consistent for use.

In contrast to constructors, for example, static initializers aren't inherited and are only executed once when the class is loaded and initialized by the JRE. In the example above, the class variable *foo* will have the value 998877 once initialization is complete.

Note also that static initializers are executed in the order in which they appear textually in the source file. Also, there are a number of restrictions on what you can't do inside one of these blocks such as no use of checked exceptions, no use of the *return* statement or the *this* and *super* keywords.

Personally, I think this is yet another odd little wart in the Java language that is due to the fact that Java doesn't have first-class classes. Sigh.

53) Are classes that implement *Serializable* required to have no-argument constructors?

No. This is a common misconception. The deserialization process does not use the object's constructor - the object is instantiated without a constructor and initialized using the serialized instance data. The only requirement on the constructor for a class that implements *Serializable* is that the first non-serializable superclass in its inheritance hierarchy must have a no-argument constructor. (See <http://www.jguru.com/jguru/faq/view.jsp?EID=34802> for a more complete explanation). This makes sense: deserialization needs to reconstruct the entire object state, which includes the state of any superclasses. If the superclass is not itself serializable, then deserialization needs to instantiate that superclass from scratch - thus the requirement. For example, with the following class:

```

public class MySerializableClass implements Serializable {
    ...
}

```

you do *not* need a no-argument constructor. MySerializableClass meets all requirements because its first non-serializable superclass, Object, has a no-argument constructor. In the following example:

```
public class MyFirstClass {  
}  
public class MySecondClass extends MyFirstClass implements Serializable {  
    ...  
}
```

MyFirstClass has the default, no-argument constructor, so no other constructor is needed. If, however, MyFirstClass defined constructors which accepted arguments without also explicitly declaring a no-argument constructor, then you would get a NotSerializableException when trying to serialize MySecondClass.

All the requirements for an object that implements Serializable are listed at <http://www.jguru.com/jguru/faq/view.jsp?EID=31434>.

54) What is the purpose of using a static method?

A static method is simply a method that does not need an instance of a class to execute.

Static methods are most useful as utility methods in classes that don't require state. For example, the java.lang.Math class has several static methods that you can call for calculations. No state is needed, so you don't need an instance of the Math class to perform the methods.

You call static methods using the *class name* rather than an instance name. For example

```
int x = Math.max(x,y);
```

Static methods are often used to access class variables. Class variables are variables in the class that are declared static. There is only *one* instance of the class variables on the system, so they make a great place to store constant values, "global" counts, and singleton instances.

Often, developers will define variables like

```
private static int count = 0;
```

and define a methods to access it, like

```
public static int getCount() {  
    return count;  
}  
public static void incrementCount() {  
    count++;  
}
```

This allows reading and modification of the class variables while retaining control of the values. Noone can directly access private class variables other than this class, so you can restrict the types of updates (for example, only incrementing the count).

Note that static methods *cannot* access any instance variables defined in the object. Why? Because you don't *have* an instance of the object to know *which* instance variables you want to access.

55) What are the differences between `System.gc()` and `Runtime.gc()`?

The methods are basically equivalent. Both suggest to the JRE that it might be a good idea if the garbage collector actually does some work now.

Specifically, `System.gc ()` is a static method so it's a wee bit more convenient to use rather than having to do e.g., `Runtime.getRuntime ().gc ()`.

56) How do I convert a `java.sql.Timestamp` to a `java.util.Date`?

While `Timestamp` extends `Date`, it stores the fractional part of the time within itself instead of within the `Date` superclass. If you need the partial seconds, you have to add them back in.

```
Date date = new Date(ts.getTime() + (ts.getNanos() / 1000000 ));
```

57) What's the difference between an interface and an abstract interface?

Nothing. All interface's are, by definition, abstract. In fact, the JLS says that use of the abstract modifier is obsolete and therefore should not be used in new programs.



58) Why is method invocation in Java polymorphic but field access isn't?

The designers of the Java programming language feel that the lack of dynamic lookup on field accesses provides greater run-time efficiency. In addition, if people want dynamic lookup on fields then they can easily use an accessor (aka "getter") instance method instead of direct field access. I.e., they leave the tradeoff up to the programmer.

59) What is a package in Java?

Java programs are organized as sets of packages. A package comprises one or more compilation units. The naming of packages is hierarchical and may comprise classes, interfaces, and sub-packages. Each package defines a namespace.

60) What is meant by compatible equals() and hashCode() methods?

In order for the Java Collections to work properly (and everything else in Java), the equals() and hashCode() methods must be compatible. Here, compatible means that if equals() reports that two instances are the same, then the hashCode() of both instances must be the same value.

61) If an object is an array, how do I find out its type?

Use the getComponentType() method of the class of the object that is the array.

```
public class ArrayReflection {
    public static void main (String args[]) {
        printType(args);
    }
    private static void printType (Object object) {
        Class type = object.getClass();
        if (type.isArray()) {
            Class elementType = type.getComponentType();
            System.out.println("Array of: " + elementType);
        }
    }
}
```

62) Does Java use *pass by value* or *pass by reference* semantics when passing parameters to methods?

Java passes parameters to methods using *pass by value* semantics. That is, a copy of the value of each of the specified arguments to the method call is passed to the method as its parameters.

Note very clearly that a lot of people confuse the "reference" terminology with respect to Java's *references to objects* and *pass by reference* calling semantics. The fact is, in dealing with Java's *references to objects*, a copy of the *reference's* value is what is passed to the method -- it's *passed by value*. This is, for example, why you cannot mutate the (original, caller's) reference to the object from inside the called method.

63) What is the difference between an API and a framework?

Here is a less formal definition:

A framework is a set of classes which handles the flow necessary to perform a complex task, but which requires plug-in classes specific to your application (database, app server), and which allows for additional plug-in classes to extend the functionality. For example, you could have a framework which handles the general flow of an online store. This framework would handle tasks such as "put item in shopping cart", "generate order on checkout", etc. On the other hand, you would need to provide the classes which enable the framework to persist the shopping cart to the database of your choice, to connect to a credit-card processor, to send orders to a warehouse, among other things. Further, if you wanted to decided that you wanted to offer monogramming, you would need to provide classes to do so, and the framework should make it easy to plug these in.

An API is really no more than a set of method signatures and other information necessary to use a class or set of classes. The API is totally distinct from the implementation. For example, a number of vendors have implemented the servlet API, each in a different way. The API by itself has no implementation.

A class library would have an API, as would a framework.

64) What are differences between type-casting and Data Conversion?

TypeCasting is a feature of the language whereby you are telling the compiler (and readers of your source code :-)) that you explicitly want it to treat the object as if it were really of the type that you specified rather than it's actual type.

Data conversion is physically changing one chunk of data into another, often with interpretation.

The difference can be illustrated thus:

Typecasting:

```
Object o = new String("1234");
```

```
String myString = (String)o;
```

This allows you to cast the type of o to it's proper type, String.

Data conversion:

```
String o = new String("1234");
```

```
Integer i = Integer.parseInt(o);
```

This converts the String "1234" into the Integer which represents the value 1234 (one thousand two hundred twenty four).

65) Why does Java not support function/method pointers/references ala C/C++?

Because Java is a more pure object-oriented language than C/C++. Function pointers and method references are ways of pulling apart the functionality of objects.

The Java, OO way of doing the equivalent is pass around an object reference where that object adheres to a particular interface and so you can invoke the appropriate methods.

### Collections

1) How do I use an Iterator to go through a Collection?

```
Collection collection = ...;
Iterator iterator = collection.iterator ();
while (iterator.hasNext()) {
    Object element = iterator.next ();
    // Do something with element
}
}
```

2) How do I use a ListIterator to go through a List backwards?

```
List list = ...;
ListIterator iterator = list.listIterator(list.size());
while (iterator.hasPrevious()) {
    Object element = iterator.previous();
    // Process element
}
}
```

3) How do I use Enumeration to iterate through a collection?

```
Enumeration enum = ...;
while (enum.hasMoreElements()) {
    Object element = iterator.nextElement();
    // process element
}
```

#### 4) How do I make an array larger?

You cannot directly make an array larger. You must make a new (larger) array and copy the original elements into it, usually with `System.arraycopy()`. If you find yourself frequently doing this, the `Vector` class does this automatically for you, as long as your arrays are not of primitive data types.

#### 5) How do you store a primitive data type within a `Vector` or other collections class?

You need to wrap the primitive data type into one of the wrapper classes found in the `java.lang` package, like `Integer`, `Float`, or `Double`, as in:

```
Integer in = new Integer (5);
```

#### 6) How do I use an array with the Collections Framework?

The `Arrays.asList ()` method provides a fixed-length `List` view of an array, where changes to the `List` are stored in the original array. The `Arrays` class also provides additional support methods for sorting and searching an array.

#### 7) Which is faster, synchronizing a `HashMap` or using a `Hashtable` for thread-safe access?

Because a synchronized `HashMap` requires an extra method call, a `Hashtable` is faster for synchronized access.

#### 8) Which is the preferred collection class to use for storing database result sets?

When retrieving database results, the best collection implementation to use is the `LinkedList`.

The benefits include:

- Retains the original retrieval order

- Has quick insertion at the head/tail

- Doesn't have an internal size limitation like a `Vector` where when the size is exceeded a new internal structure is created (or you have to find out size beforehand to size properly)

- Permits user-controlled synchronization unlike the pre-Collections `Vector` which is always synchronized

Basically:

```
ResultSet result = stmt.executeQuery("...");
List list = new LinkedList();
while(result.next()) {
    list.add(result.getString("col"));
}
```

If there are multiple columns in the result set, you'll have to combine them into their own data structure for each row. Arrays work well for that as you know the size, though a custom class might be best so you can convert the contents to the proper type when extracting from database, instead of later.

Why doesn't the Iterator interface extend the Enumeration interface?

If the Iterator interface extended the Enumeration interface, the Iterator interface would end up with five methods where two methods just called other methods in the interface. The designers of the framework wanted to get rid of the cumbersome Enumeration method names so had the Iterator interface stand on its own with new shorter method names.

9) How do I print a Collection?

The Collection Framework implementation classes override the toString() method to print out all the elements of the collection. If you create your own custom implementation, as long as your class is a subclass of AbstractMap or AbstractCollection you'll inherit this behavior. (Keep in mind that ArrayList and AbstractSet subclass AbstractCollection.)

10) How do I synchronize a collection?

With the Collections Framework, the new implementations are all unsynchronized by default. If you need synchronized access, you must synchronize things yourself. The Collections class offers a wrapper method for each of the six core collection interfaces that add synchronization to an arbitrary collections implementation. To ensure thread-safety, direct access to the original backing collection must be avoided.

For example, the following will synchronize an arbitrary List and lose the original reference so you can't access it directly:

```
List list = ...;  
list = Collections.synchronizedList(list);
```

11) How do I get the length of an array?

To avoid getting an ArrayIndexOutOfBoundsException, you can check for the array length from either the length instance variable or using reflection and calling java.lang.reflect.Array.getLength(), passing the array as an argument to the method.

```
int length = args.length;  
// or  
int length2 = Array.getLength(args);
```

12) How do I sort an array?

The Arrays class in java.util provides a series of sort () methods for sorting arrays. If the array is an array of primitives or an array of a class that implements Comparable then you can just call the method directly:

```
Arrays.sort(theArray);
```

If, however, it is an array of objects that don't implement the Comparable interface then you need to provide a custom Comparator to help you sort the elements in the array.

```
Arrays.sort (theArray, theComparator);
```

13) What's the fastest way to traverse all the elements of a Vector?

If speed is of the essence, do not use a Vector. Instead, use an ArrayList. All accesses to a Vector are synchronized, whether you need it or not. If you know you aren't accessing the structure from multiple threads, the ArrayList will be much faster.

With that said, what about if you need to stay with a Vector? There are at least four ways to traverse through all the elements of a Vector:

Using a for loop and accessing each element with `elementAt(index)` or `get(index)`

Using an Enumeration

Using an Iterator

Using an Enumeration/Iterator, but relying on an exception to catch ending

Of the four, there are negligible differences in performance. While looping through to a specific index is a little faster, you lose the flexibility of changing to another data structure at a later time. With the Enumeration / Iterator approach, there is the added cost to create the Enumeration/Iterator to walk through. With the exception approach, you are relying on `NoSuchElementException` to be thrown to catch the ending of the walk through. While creating the exception takes time, this approach avoids the repeated test of `hasMoreElement()`. As the structure gets larger, avoiding the test condition can save lots of time.

14) How does a Hashtable internally maintain the key-value pairs?

The Hashtable class uses an internal (private) class named `Entry` to hold the key-value pairs. All entries of the Hashtable are stored in an array of `Entry` objects with the hash value of the key serving as the index. If two or more different keys have the same hash value these entries are stored as a linked list under the same index.

15) How do I look through each element of a HashMap?

To go through all the elements of a `HashMap`, or any class that implements the `Map` interface, call the `entrySet()` or `keySet()` methods than loop through what is returned. The `entrySet()` returns a group of `Map.Entry` elements, whereas the `keySet()` method just returns a `Set` of key elements. If you then want what the key refers to, you'd have to look them up.

Once you have a `Set` to work with, you would then use an `Iterator` to go through all its elements. The following demonstrates:

```
Map map = some hash map
Set set = map.keySet();
Iterator it = set.iterator();
while (it.hasNext()) {
    System.out.println(it.next());
}
```

16) How do I create a read-only collection?

The `Collections` class has six methods to help out here:

`unmodifiableCollection (Collection c)`

`unmodifiableList (List list)`

`unmodifiableMap (Map m)`

`UnmodifiableSet (Set s)`

`unmodifiableSortedMap (SortedMap m)`

`unmodifiableSortedSet (SortedSet s)`

If you then get an `Iterator` from one of these unmodifiable collections, when you call `remove()` it will throw an `UnsupportedOperationException`

17) How can I process through the keys of a Hashtable in sorted order?

In order to get all the keys for a Hashtable, you use the `keys()` method to get an Enumeration or the `keySet()` method to get a Set. If you are using Java 2, and can use the collections framework, what you should do is get the key set of the Hashtable and create a TreeSet from it. You can then get an iterator() from the created TreeSet that will have the keys in order. If you can't use the collections framework, you'll have to sort the Enumeration you get back from `keys()` yourself.

18) Which collections in Java are synchronized and which aren't?

The original collection classes in Java are all synchronized: Vector and Hashtable, along with their subclasses Stack and Properties. Those classes introduced with the Java 2 Collections Framework are all NOT synchronized by default, the sets, lists, and maps. If you need to synchronize those, see [How do I synchronize a collection?](#).

19) What are the differences between ArrayList and LinkedList?

An ArrayList is a List implementation backed by a Java array, similar to the Vector class. As the number of elements in the collection increases, the internal array grows to fit them. If there are lots of growth periods, performance degrades as the old array needs to be copied into the new array. However, random access is very quick as it uses an array index to access. With a LinkedList, the List implementation is backed by a doubly linked list data structure, allowing easy inserts/deletions anywhere in the structure, but really slow random accesses as the access must start at an end to get to the specific position.

20) What is the difference between a singly linked list and doubly linked list?

A singly linked list is one that has only a pointer/reference to the next element in the list. A doubly linked list has both a previous and next pointer/reference.

21) What is a polymorphic algorithm?

In general, an algorithm is called polymorphic if it can achieve the same functionality using different data structures. For example, if the same sorting or searching algorithm could be used with different types of containers, such as vectors, lists, maps, etc., then that algorithm would be called polymorphic. As a case in point, the generic algorithms of the STL library in C++ are polymorphic because they can be used for different container classes.

In the context of Java collections, the polymorphic algorithms are all supplied by the various static methods of the `java.util.Collections` class. Consider, for example, the method `sort( List list )`. This method is capable of sorting any collection that has implemented the List interface, and that includes containers of types ArrayList, LinkedList, Vector, etc.

22) How do you sort the elements of a vector?

Since the Vector class implements the List interface, you can call the `Collections.sort()` method to sort the elements in place. You can also manually insert elements into a Vector to keep the vector sorted. Of course, if keeping sorted access is such a big deal, you should consider using a different, more appropriate data structure like a TreeSet.

For 1.1 Java users, you'll need to sort the elements yourself. There is no built-in support for this. A better option might be to sort the elements as you insert each element.

23) How do you control growth of vectors when their internal arrays are full?

The vector constructor can include either an initial capacity or a capacity and growth increment. When not specified, the initial size of the vector is 10 and growth will double when necessary. Otherwise, initialize size and growth will grow when needed as specified by the arguments to the constructor.

If the argument to the constructor is a collection, the initial size of the internal structure is 10% larger than the collection. Since there is no second argument to control growth, the capacity will double when necessary.

24) Why can't I add a collection to itself?

This will cause a stack overflow exception to be generated on calls to methods like `toString()` and `hashCode()`, which recursively call the method on the elements of the collection.

25) What is a weak reference and what are they used for?

Normally the Java garbage collector plays safe. It will only free up the memory used by an object when that object can no longer be accessed by the program. Once an object become impossible to reach it is eligible for collection, and eventually its memory will be reclaimed. This eliminates one of the most common programming errors in some other languages (like C++), where code accidentally tries to access an object that has been freed. Unfortunately it can lead to another problem, where you leave open a potential access route to an object that you don't need any more. Memory fills up, and the program slows down or reports an "Out of Memory" error.

To avoid this, you can be very careful to close off access paths to an object once you have finished using it. Java 2 introduces another alternative, the weak reference. Weak references provide access to an object without preventing it from being freed. When you use a weak reference you have to accept that the object referred to may have disappeared, which results in the reference being automatically set to null. On the other hand, the weak reference will not hold the object in memory once it is inaccessible via normal references (or via "soft" references - see below). Weak references are not appropriate in all circumstances, but sometimes they can make code easier to write and understand.

The most common use of weak references is indirect - they are used internally by the `WeakHashMap` class. Like `HashMap`, `WeakHashMap` associates key objects with values. However, once the key object becomes inaccessible via stronger references it becomes eligible for garbage collection. When it is freed, the map entry magically disappears. The assumption here is that if you are not using the key anywhere other than in the map you will have no need to look it up, so it should be freed.

Other specialist references are soft references (which inhibit collection until memory runs short), and phantom references (used for cleanup when objects are freed).

For more detailed (and precise) information, see the `java.lang.ref` API docs, and also the article [Reference Objects and Garbage Collection](#) at the Sun website.

26) How does `ArrayList` increase its capacity?

Unlike `Vector` where you can specify a capacity increment, `ArrayList` doesn't support this. Instead, `ArrayList` will increase capacity by about a half when it runs out of space. The reference implementation uses the formula:

$$\text{newCapacity} = (\text{oldCapacity} * 3) / 2 + 1$$



though, this isn't part of the class definition so others can implement it differently.

27) What are the differences between HashMap and Hashtable?

Both provide key-value access to data. The Hashtable is one of the original collection classes in Java. HashMap is part of the new Collections Framework, added with Java 2, v1.2. The key difference between the two is that access to the Hashtable is synchronized on the table while access to the HashMap isn't. You can add it, but it isn't there by default.

Another difference is that iterator in the HashMap is fail-safe while the enumerator for the Hashtable isn't. If you change the map while iterating, you'll know.

And, a third difference is that HashMap permits null values in it, while Hashtable doesn't.

For new code, I would tend to always use HashMap.

28) What are the differences between Vector and ArrayList? Which is best to use?

Vector and ArrayList are very similar. Both of them represent a 'growable array', where you access to the elements in it through an index.

ArrayList it's part of the Java Collection Framework, and has been added with version 1.2, while Vector it's an object that is present since the first version of the JDK. Vector, anyway, has been retrofitted to implement the List interface.

The main difference is that Vector it's a synchronized object, while ArrayList it's not.

While the iterator that are returned by both classes are fail-fast (they cleanly thrown a ConcurrentModificationException when the original object has been modified), the Enumeration returned by Vector are not.

Unless you have strong reason to use a Vector, the suggestion is to use the ArrayList.

## Threads

1) What is a thread?

A thread is a set of instructions executing apart from other threads (with its own stack) but sharing the same memory space (with the same heap).

2) How do I create a new thread and have it start running?

Creating a thread involves creating a new Thread and invoking its start() method. Calling start() causes the run() method of the Thread subclass or the Runnable object passed to the Thread constructor to execute.

```
Thread t1 = new Thread() {
```

```

    public void run() {
        for (int i=0; i<100; i++) {
            System.out.println("Tastes Great");
        }
    }
};

```

```

Runnable r = new Runnable() {
    public void run() {
        for (int i=0; i<100; i++) {
            System.out.println("Less Filling");
        }
    }
};
Thread t2 = new Thread(r);

```

```

t1.start();
t2.start();

```

### 3) How do I get a thread to pause?

The static `sleep()` method of the `Thread` class will let your thread sleep for a set number of milliseconds (or nanoseconds). When the thread wakes up, it will be scheduled to execute in the future. It may not start executing immediately after the timeout.

```

try {
    Thread.sleep(3000); // 3 seconds
} catch (InterruptedException e) {
    System.err.println("You interrupted me");
}

```

[

You can have another thread wake up the sleeping thread prematurely by calling `t.interrupt()` (where "t" is a pointer to the thread object).

Note that since `Thread.sleep` is a static method, the following code will not do what you expect it to:

```

Thread t = new MyThread();
t.start();
try {
    t.sleep(3000);
} catch (InterruptedException e) {

```

```
    System.err.println("You interrupted me");  
}
```

It will pause the current thread, meaning the thread executing the shown code, not the child thread (t/MyThread).

Non-static synchronized methods synchronize on the instance (this) of the class.

4) What object does static synchronized methods use for locking?

Static synchronized methods synchronize on the class object (this.getClass()) of the class.

5) How can one thread wait for another thread to die before it continues execution?

The thread's join() method allows you to wait for another thread to finish execution.

```
Thread t1 = new Thread(runnable);  
t1.start();  
// do stuff  
...  
// wait for t1 to finish  
t1.join()
```

6) What is the use of start () function in starting a thread? Why we do not use the run() function directly to run the thread?

The start() method tells the Java Virtual Machine that it needs to create a system specific thread. After creating the system resource, it passes the Runnable object to it to execute its run () method. Calling the run() method directly has the "Thread" execute in the same thread as the calling object, not in a separate thread of execution, as intended.

In other words, calling run() is just like a normal method call. Whereas, calling start() tells the thread to create the necessary system resources and then execute the run() method asynchronously.

7) What is the main difference between a preemptive scheduler and a non-preemptive scheduler?

A preemptive scheduler interrupts a thread of execution when its timeslice runs out. A non-preemptive (or "cooperative") scheduler waits for the thread to yield control.

[Java native thread implementations are usually preemptive; the green-threads implementation is cooperative but priority-preemptive, which means that when a high-priority thread becomes runnable, it immediately becomes active. -Alex]

8) How does a preemptive scheduler manage a thread's timeslice?

The lowest level preemptive scheduler (kernel layer) uses the system timer interrupt and context switching to manage timeslices. When any CPU interrupt occurs, the CPU makes a note of where it was, and what it was doing (pushes all registers onto the stack). The CPU then processes the interrupt and returns to what it was previously doing (pops all registers from the stack). The thread context in this sense, is the information the CPU needs to start or resume execution in any section of code. The scheduler is invoked by the timer interrupt routine (it can also be part of the timer interrupt). The scheduler checks to see if the current timeslice has expired; if so, the current thread context is stored and the next valid thread context is restored. The most basic implementation is a stack swap, as each thread has its own stack. When a thread is created, it gets a new context with an empty stack. The new context directs the CPU to the thread's `run()` member at the beginning of its timeslice. A thread's context is destroyed when the thread returns from the `run()` member, or its `stop()` member is successfully invoked.

9) What are the differences between extending the Thread class and implementing the Runnable interface?

Extending the Thread class will make your class unable to extend other classes, because of the single inheritance feature in JAVA. However, this will give you a simpler code structure. If you implement runnable, you can gain better object-oriented design and consistency and also avoid the single inheritance problems.

10) What are the different uses of the synchronized keyword?

The keyword synchronized is used to acquire a exclusive monitor lock on an object. It can be used to mark either a method or a block of code. In both cases, it means to acquire a lock for the duration of the method or block, and to release the lock at the end. It also takes a parameter, which names the object whose lock

will be acquired. (The parameter is implicit when marking a method, as shown below.)

```
synchronized (foo) {  
    ...  
}
```

Acquires a lock on the object instance "foo" at the open brace, and releases the lock at the close brace.

```
synchronized (this) {  
    ...  
}
```

Acquires a lock on the current object instance ("this") at the open brace, and releases the lock at the close brace.

```
synchronized void bar() {  
    ...  
}
```

Acquires a lock on the current object instance at the open brace, and releases it at the close brace. This is equivalent (\*) to

```
void bar() {  
    synchronized (this) {  
        ...  
    }  
}
```

```
class Foo {  
    synchronized static void bar() {  
        ...  
    }  
}
```

Acquires and releases a lock on the class instance of class Foo. Every class, when loaded, is given an instance of class Class. That means that no matter who invokes method Foo.bar(), the lock will be on the static instance, and not on any specific instance of class Foo.

I know this sounds confusing, but it has the same semantics as any other use of static: all statics (methods, variables, etc) are

essentially global, interact with all other statics of the same class, and do not interact with non-static instance data.

Whether a method is public or not makes no difference to the semantics of synchronized.

11) How can multiple threads be controlled simultaneously?

If you create threads in a ThreadGroup object, they may be controlled simultaneously with the member functions of said object.

12) What is a daemon thread? When should I use setDaemon() and why?

Use thread.setDaemon(true) to tell the JVM to make the thread a daemon thread.

According to Webster's, a daemon (variant of demon) is an attendant power or spirit. Daemon threads are typically used to perform services for your application/applet (such as loading the "fiddley bits"). The core difference between user threads and daemon threads is that the JVM will only shut down a program when all user threads have terminated. Daemon threads are terminated by the JVM when there are no longer any user threads running, including the main thread of execution. Use daemons as the minions they are.

[In short: daemon threads do not keep the program from quitting; user threads keep the program from quitting. -Alex]

13) What is the difference between sleep(), wait() and suspend()?

Thread.sleep() sends the current thread into the "Not Runnable" state for some amount of time. The thread keeps the monitors it has acquired -- i.e. if the thread is currently in a synchronized block or method no other thread can enter this block or method. If another thread calls t.interrupt() it will wake up the sleeping thread.

Note that sleep is a static method, which means that it always affects the current thread (the one that is executing the sleep method). A common mistake is to call t.sleep() where t is a different thread; even then, it is the current thread that will sleep, not the t thread.

t.suspend() is deprecated. Using it is possible to halt a thread other than the current thread. A suspended thread keeps all its monitors and since this state is not interruptable it is deadlock prone.

`object.wait()` sends the current thread into the "Not Runnable" state, like `sleep()`, but with a twist. Wait is called on a object, not a thread; we call this object the "lock object." Before `lock.wait()` is called, the current thread must synchronize on the lock object; `wait()` then releases this lock, and adds the thread to the "wait list" associated with the lock. Later, another thread can synchronize on the same lock object and call `lock.notify()`. This wakes up the original, waiting thread. Basically, `wait()/notify()` is like `sleep()/interrupt()`, only the active thread does not need a direct pointer to the sleeping thread, but only to the shared lock object.

[This answer was edited; the original answer was clear but I felt I should expand on some points; please blame me, not Ingo, for any errors. -Alex]

14) What is the difference between a lightweight and a heavyweight process?

[Short answer: threads are lightweight, programs (aka processes or tasks) are heavyweight. -Alex]

Lightweight and heavyweight processes refer the mechanics of a multi-processing system.

In a lightweight process, threads are used to divvy up the workload. Here you would see one process executing in the OS (for this application or service.) This process would possess 1 or more threads. Each of the threads in this process shares the same address space. Because threads share their address space, communication between the threads is simple and efficient. Each thread could be compared to a process in a heavyweight scenario.

In a heavyweight process, new processes are created to perform the work in parallel. Here (for the same application or service), you would see multiple processes running. Each heavyweight process contains its own address space. Communication between these processes would involve additional communications mechanisms such as sockets or pipes.

The benefits of a lightweight process come from the conservation of resources. Since threads use the same code section, data section and OS resources, less overall resources are used. The drawback is now you have to ensure your system is thread-safe. You have to make sure the threads don't step on each other. Fortunately, Java provides the necessary tools to allow you to do this.

15) What is the meaning of calling a method or object "thread-safe"?

Basically, calling a method "thread-safe" means that even if multiple threads try to access it simultaneously, nothing bad happens. Here "bad" usually means that due to race conditions, or deadlock, or other pitfalls, the object's state gets corrupted, or its methods produce unreliable results. A method usually achieves

thread-safety by protecting access to shared resources. This usually translates to using the Java synchronized keyword to protect blocks of code that access instance variables, or other shared variables.

For an object to be thread-safe, it must be possible for multiple threads to simultaneously access the same method, or multiple methods, in that object. Usually this is achieved by assuring that each method is thread-safe, but this doesn't always suffice, since methods can call each other in strange ways, leading to deadlock and other weirdness.

It is very difficult to prove that an object is thread-safe. The main rule of thumb for making thread-safe objects is, "Make all the instance variables private, and all the public accessor methods synchronized." However, this is sometimes difficult to achieve in practice, due to exigencies of performance, architecture, or implementation.

Accurate multithreaded programming is a true art, and very difficult to master. Read "Java Threads" by Oaks and Wong, and "Concurrent Programming in Java" by Lea, for inspiration in your quest to become a thread-safe programmer.

16) How can I actually, really deallocate a Thread to release the memory? Setting thread = null does not work!

Using thread = null will not release a running thread. In order to release the memory associated with a thread, you need to make sure that all of the following are done:

- Make sure that the thread's start() method has been called.
- Make sure that the thread has stopped executing.
- Clear any references to that Thread object (thread = null;).

This is the best you can do to ensure the release of memory for a Thread. You have to call start() on the thread because several JVMs have a bug where they will not release all the thread's memory if the thread is not started.

17) What is the difference between a thread and a process?

A process is an OS-level task or service. A thread runs "inside" a process and may be virtual or simulated. Generally speaking, threads share resources like memory, where processes each have their own separate memory area, and need to take more elaborate steps to share resources.

Another name for thread is "lightweight process" to distinguish it from the "heavyweight" system processes.



18) What is the difference between threads and interrupts ?

A thread is a CPU's state of execution as it processes a set of instructions (also referred to as a task). An interrupt is a condition that causes the CPU to store the state of its current thread of execution to begin a more important task, or to begin or resume the next task in a list of tasks. An interrupt handler is the set of CPU instructions associated with any given interrupt (a PC has several types of interrupts).

The confusing part is the fact that the thread of execution in an interrupt handler is often referred to as an interrupt.

In short, a thread is a task and an interrupt is a signal used to queue a more important task.

19) When and why is `IllegalMonitorStateException` thrown?

According to the JavaDoc, `IllegalMonitorStateException` is thrown "to indicate that a thread has attempted to wait on an object's monitor or to notify other threads waiting on an object's monitor without owning the specified monitor."

20) What is deadlock? How can I eliminate it?

A simple deadlock situation is one in which a two threads are waiting. Each thread waiting for a resource which is held by the other waiting thread. [In Java, this resource is usually the object lock obtained by the `synchronized` keyword.]

This deadlock situation can range from the above two thread situation to any number of threads waiting in a circular fashion.

Many solutions to this well known problem have been defined over the years including the large range of solutions to "The Dining Philosophers" problem.

21) What is `InterruptedException`? Why do we need to catch it when calling `Thread.sleep()`?

`InterruptedException` is thrown if another thread calls `interrupt()` on the sleeping thread while it is asleep. If `sleep()` is setting an alarm clock before bed, `interrupt()` is a midnight phone call.

This means that you can safely ignore the exception (with `catch (InterruptedException e) {}`) since in general, only code that you write will ever call `interrupt`. Since you didn't write it, you know it won't happen. :-)

InterruptedException was not implemented in Java 1.0; however, it is definitely available now.

The exception will also get thrown if the "interrupted" flag is set in the thread, which happens if another thread calls interrupt even before the original thread sleeps.

22) When exactly is the AWT thread started?

Anytime the java.awt.Toolkit instance is (directly or indirectly) created. The direct way to create it is by calling-

```
Toolkit.getDefaultToolkit();
```

Indirect way is by creating any AWT object.

Here is a sample program to play with -

```
import java.awt.*;
public class X
{
    static Frame f;

    public static void main(String[] args)
    {
        //Toolkit.getDefaultToolkit();
        f = new Frame();
        System.out.println("...");
    }
}
```

23) What is the difference between multithreading and multitasking? What about multiprogramming? Multiprocessing?

Multitasking is running multiple "heavyweight" processes (tasks) by a single OS.

Multithreading is running multiple "lightweight" processes (threads of execution) in a single process / task / program. See [What is the difference between a lightweight and a heavyweight process?](#) for more detail on lightweight vs. heavyweight processes.

Multiprogramming is essentially a synonym for multitasking (though multitasking connotes sharing more resources than just the CPU, and is the more popular term).

Multiprocessing involves using multiple CPUs, either in the same (SMP) or different (MPP) host boxes, to run a program. See [whatis.com](http://whatis.com) for a good definition.

Most Java implementations will split threads among different processors if they're running on an SMP box.

24) Why is Thread.run() declared public? Shouldn't it be declared protected, as only the Thread superclass can invoke it?

One reason is that Thread implements Runnable, which requires that run() be public. This is an unfortunate consequence of the definition of Java interfaces.

It is also not clear whether you would want to prohibit other classes from calling a Thread's run() method in a synchronous manner -- as it is, they have the option of calling either run() or start() depending on their needs.

25) Is there any method by which I can join() to a group of threads instead of a single one?

Yes. Just join to each one in turn. That way, when the loop exits, you know that all the threads have exited -- whether it was the first thread or the seventh thread that took the longest, all threads will be waited for. Remember, join() on a thread that's already exited takes no time.

```
Iterator i = myThreads.iterator();
while (i.hasNext()) {
    ((Thread)i.next()).join();
}
```

26) Why do I have to call System.exit() from my main method?

Unfortunately, certain system threads are started as regular threads, not [daemon threads](#). One example is the [AWT thread](#). That means that if your app opens a window, then even if that window is closed, the AWT thread continues, so your app will never quit on its own.

There are many cases where a Java program will quit on its own when main() exits, but there are also many sneaky cases where a thread is launched without your knowledge by a library routine. Database Connection Pools are notorious for doing this.

A warning: Make sure you only call System.exit() when you're really really sure all background threads have stopped processing. Otherwise the threads will be

stopped prematurely, and may leave things in a bad state (for instance, if they're halfway through writing a file).

27) What is the difference between sleep and yield?

`yield()` tells the JVM Thread Scheduler that it's OK to give other threads time slices. Usually the JVM uses this call to activate another thread of the same thread priority. In a good preemptive multithreading environment, `yield()` is a no-op. However, it is important in a cooperative multithreading environment, since without `yield()`, one thread can eat up all of the CPU.

`sleep(x)` tells the JVM Thread Scheduler to actively put this thread to sleep and not run it again until at least x milliseconds have elapsed.

Neither `sleep()` nor `yield()` change anything about the status of synchronization locks. If your thread has a lock, and you call `sleep(1000)`, then at least a second will elapse before your thread wakes up. When it wakes up it may decide to release the lock -- or it may hold on to it longer.

28) How do we stop the AWT Thread? Our Swing application will not terminate.

You need to explicitly call `System.exit(exit_value)` to exit a Swing application. This is because the event dispatcher thread is not a Daemon thread, and won't allow the JVM to shut down when other threads are dead.

29) How soon after calling `start()` will the `run()` method be executed?

Your `run` method is not guaranteed to run immediately after the `Thread.start()` method is called.

The Java Language Specification gives implementations lots of legroom with respect to the scheduling of threads. Your thread may start immediately, or it may start 5 minutes later.

Any decent implementation will have your thread start as soon as possible under the scheduling algorithm, so a five minute wait would be unreasonable.

30) What does it mean to lock an object?

Every object instance in Java has a little piece of data hanging off of it called a "monitor lock." This is similar to a [semaphore](#).

When you use the [synchronized](#) keyword, the current thread attempts to obtain the monitor lock for the object in question (either "this" or the object named

explicitly). If the lock is unavailable, because another thread has acquired it, then the current thread pauses indefinitely. When it wakes up again, it will have acquired the lock, and will retain it until the end of the code block where synchronized was used.

One confusing part is that the only thing that this lock locks is access to the lock itself. It does not control access to instance variables or method access in and of itself. Other threads can reach in and modify variables and call methods. If you want the lock to control access to variables, then you must explicitly use the synchronized keyword every single place in your code that accesses them.

A straightforward way to guarantee that no variable access happens without a lock is by making all your variables private and using synchronized on all your accessor/mutator methods (getFoo/setFoo).

31) How do I capture an exception stack trace and put it into a string?

Here's how to print the trace in a string:

```
Exception ex = new Exception("something went wrong");
StringWriter sw = new StringWriter();
ex.printStackTrace(new PrintWriter(sw));
String stacktrace = sw.toString();
System.out.println("stacktrace = " + stacktrace);
```

32) Is it possible to wake up a sleeping thread?

Call `sleepingThread.interrupt()`.

## I/O

1) How can I compress my data to save bandwidth when sending across a socket?

The GZIPInputStream and GZIPOutputStream classes found in the java.util.zip package compress data with the GZIP file format, as defined by RFC 1952

2) How do I append to end of a file in Java?

You can use java.io.RandomAccessFile and something like the following:

```
try {
    RandomAccessFile raf =
        new RandomAccessFile("filename.txt", "rw");
    raf.skipBytes( (int)raf.length() );
    // You are now at the end of the file,
    // and can start writing new data out, e.g.
    raf.writeBytes(
        "Log restarted at 13:00pm 3-2-2000\n");
    raf.close();
} catch (IOException ex ) {
    ex.printStackTrace();
}
```

or you can use FileWriter / FileOutputStream and open it for append:

```
FileWriter writer =
    new FileWriter("filename.txt", true);
```

3) How can I see from an applet the names of files and directories that are in the server?

There is no built-in support for this. Basically, you must create a service on the server that when requested returns a list of available files and directories.

4) How can I read .zip and .jar file using standard Java classes?

The ZIP and JAR reading classes are found in the java.util.zip and java.util.jar packages respectively. The following demonstrates reading from a ZIP file, listing all the files in the zip and displaying the contents of the first file in the zip. JAR file reading is similar, just with different classes and having a manifest.

```
import java.util.zip.*;
import java.util.*;
```

```

import java.io.*;

public class ZipExample {
    public static void main(String args[]) {
        try {
            ZipFile zf = new ZipFile("the.zip");
            Enumeration entries = zf.entries();
            String first = null;
            while (entries.hasMoreElements()) {
                ZipEntry ze = (ZipEntry)entries.nextElement();
                System.out.println("Entry " + ze.getName());
                if (first == null) first = ze.getName();
            }
            ZipEntry ze = zf.getEntry(first);
            if (ze != null) {
                BufferedReader br = new BufferedReader(
                    new InputStreamReader(zf.getInputStream(ze)));
                long size = ze.getSize();
                if (size > 0) {
                    System.out.println(first + " Length is " + size);
                    String line;
                    while ((line = br.readLine()) != null) {
                        System.out.println(line);
                    }
                    br.close();
                }
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

```

5) How do I check for end-of-file when reading from a stream?

Exactly how depends upon which stream you are reading from. If you are reading with the `read()` method of `InputStream/Reader`, this will return -1 on EOF. If however you are using `BufferedReader.readLine()`, then this will return null on end of file. And, the `readXXX` operations of the `DataInput` interface throw an `EOFException`. If you are unclear how to terminate your reading, check the javadoc for the stream you are using.

6) Under what circumstances would I use random access I/O over sequential, buffered I/O?

Whether you use random access I/O or sequential access really depends upon what you are trying to do. Random access I/O is usually used for fixed-size data records, where you want to overwrite the original record with changes, or to create something like a rolling log file. Nowadays, there are lightweight database systems that would do this type of operation for you, adding capabilities like querying and a standard JDBC access so you wouldn't have to waste your time redesigning the wheel. While not trying to tell you to never use random access I/O, the Java `RandomAccessFile` class lives outside the Java streams class hierarchy meaning that you can't add a facade around the random access file to buffer it or enrich its capabilities in any manner. And you would also need to program in things like simultaneous access.

Sequential access is just for that, when you need to access a file sequentially from start to finish. While you can `skip()` around the data, it is generally meant to be read from beginning to end, where the whole file has some meaning. For performance reasons, it is best to always buffer your I/O, at least when using `Reader`, `Writer`, `InputStream`, and `OutputStream` classes, but not `RandomAccessFile`.

#### 7) How do I get a listing of the files in a directory?

Create a `java.io.File` for the directory, then ask for the list of files with one of the following:

```
public java.lang.String[] list();  
public java.lang.String[] list(java.io FilenameFilter);  
public java.io.File[] listFiles();  
public java.io.File[] listFiles(java.io FilenameFilter);  
public java.io.File[] listFiles(java.io FileFilter);
```

The first two return the filenames as strings (for that directory), the latter three return actual `File` objects.

#### 8) Where can I store temporary files?

The `java.io.tmpdir` system property defines an appropriate area:

```
String tempPath = System.getProperty("java.io.tmpdir");  
File f = new File(tempPath, "test.out");
```

#### 9) How can I change time and date of a file?



Provided you have write access to the file and are using JDK 1.2 or later, the public boolean `setLastModified(long time)` method of `File` allows you to modify the timestamp associated with a file.

#### 10) How do I create a temporary file?

To create a temporary file, use the `createTempFile()` method of the `File` class. This allows you to set the file prefix, suffix, and directory. If no directory is specified, then the `java.io.tmpdir` System property is used. To ensure the file is deleted when the program ends (assuming normal termination), be sure to call the `deleteOnExit()` method of the `File` object created by `createTempFile()`.

```
File temp = File.createTempFile("jguru", ".tmp");
temp.deleteOnExit();
// use temp like any other File
```

#### 11) How do I delete a file / directory?

Use the `delete()` method of the `File` class.

#### 12) How do I read text from standard input?

`System.in` is the `InputStream` for standard input. The following demonstrating reading from it a line at a time:

```
InputStreamReader isr = new InputStreamReader(System.in);
BufferedReader reader = new BufferedReader(isr);
String line = null;
while ((line = reader.readLine()) != null) {
    // Process line
}
```

#### 13) How do I write text to a file?

Writing text involves writing strings, so you would use a `FileWriter`.

```
FileWriter fw = new FileWriter(filename);
PrintWriter pw = new PrintWriter(fw);
pw.println("Hello, World");
pw.close();
```

#### 14) How do I copy a file?

To make a copy of a file, open the source file as a `FileInputStream`. Open the destination as a `FileOutputStream`. As you read a byte from the input, write it to the output. There is no built-in method to make a file copy for you.

15) What is a stream?

Basically, a stream is an ordered look at a sequence of bytes for input or output.

Low-level streams provide direct access to the underlying bytes, like a `FileInputStream`, `FileOutputStream`, or `CharArrayWriter`, where reading and writing work directly with the underlying input/output device. High-level streams, or filters, instead build upon the low-level streams to provide additional capabilities, like buffering, counting lines, compressing data, or reading/writing higher-level data members like primitives or objects. You can chain multiple filters together to get multiple higher-level operations on a single low-level stream.

16) How can I open the same file for reading as well as writing?

The `RandomAccessFile` class supports simultaneous reading and writing from the same file. Just open it in "rw" mode:

```
RandomAccessFile raf =  
    new RandomAccessFile("filename.txt", "rw");
```

Use `skipBytes()` to move to where you wish to read/write.

17) How do I list all drives/filesystem roots on my system?

The `listRoots()` method of the `File` class was introduced with the 1.2 release for this:

```
File[] roots = File.listRoots();  
for(int i=0;i<roots.length;i++)  
    System.out.println("Root["+i+"]:" + roots[i]);
```

18) How to insert content into the middle of a file without overwriting the existing content?

There is no direct support for inserting content in the middle of a file/stream. What you need to do is copy the original content into another file/stream with the new content added where necessary.

19) How do I get the creation date and time of a file?

There is no support for getting the creation date and time of a file from Java. All you can get is when the file was last modified, with the `lastModified()` method of `File`.

20) How can you combine multiple input streams to be treated as one?

The `SequenceInputStream` class in the standard `java.io` package allows you to combine multiple input streams into one. You can either create a vector of streams, passing the `elements()` to the constructor, or combine two directly in a constructor.

21) How can I rename a file?

The `File` class has a `renameTo()` method that allows you to rename files, just pass in an argument of the new name as a `File` object. A boolean status is returned to report success or failure.

22) When do you use the `Reader/Writer` classes, instead of the `InputStream/OutputStream` classes?

The `InputStream/OutputStream` classes are for reading and writing bytes, while the `Reader/Writer` classes are for reading characters / text. If you need to process text, you use `Reader/Writer`. If you need to process content at the byte level, either as the raw bytes, or as higher level data like primitives (through `DataInput/Output`), objects (through `ObjectInput/Output`), or possibly compressed data (`GZIPInput/Output`), then you would work with an `InputStrem/OutputStream`.

23) How can I make a file writable that is currently read-only?

There is only support in Java to make a writable file read-only. There is no way to go the other way. I guess someone considered this a security risk so didn't add an appropriate API to perform.

24) How can I trap system-specific key sequences like `Ctrl-Alt-Del`?

Alas, using pure, portable Java, you cannot trap those sorts of system-specific key sequences.

25) How can I get the current working directory with Java?

The current working directory is stored in the system property `"user.dir"`. The following example shows how to read this system property from your application:

```

public class userdir {
    public static void main(String[] args) {
        System.out.println("Working Directory = " +
            System.getProperty("user.dir"));
    }
}

```

26) What is a resource leak?

Garbage collection manages only *memory*, not other system resources. If your Java program has plenty of free memory, garbage collection will not be triggered automatically. Usually, however, there are other resources that are more limited than memory. For example, all OSes have limits on the number of sockets, file handles, etc. that can be open. Sometimes this limit is quite low. This is true even on a desktop, e.g. if your system has 128KB of memory, your Java program can easily allocate all the available file handles without coming near to filling up the heap. If this happens, your Java program will fail. This is what we call a *resource leak*; the unintentional maintenance of references to non-memory resources.

This is why it is important in Java to explicitly manage non-memory resources. Classes which utilize non-memory resources should provide ways to explicitly allocate/deallocate those resources, independent of garbage collection. For example Socket, InputStream and OutputStream each provide explicit close() methods for deallocation of file descriptors, Window provides a dispose() method to free the window handle, etc. The way to properly use these classes is to allocate using the constructor, then deallocate using the appropriate method (deallocation is preferably done in a finally{} block, so it will execute whether or not an exception is thrown during use). These classes *do* release these non-memory resources in their finalize() method, but remember that the finalizer only gets called by the garbage collector, and if the object is never collected, it will never be finalized, hence will never release the resources.

27) Why are there two type of I/O in Java, namely byte streams and character (Reader/Writer) streams?

The Reader and Writer classes were added to JDK 1.1 to support internationalization, since the existing streams at that time didn't properly support the use of multi-byte Unicode characters or character encodings other than ASCII. The Reader and Writer classes make it possible to work with internationalized character streams rather than byte streams.

28) Is there an easy way of counting line numbers? Or do you have to go through the entire file?

You have to go through the entire file.

```

try {
    LineNumberReader lnr = new LineNumberReader(new FileReader(filename));
    String s;
    while ((s = lnr.readLine()) != null);
    System.out.println("Lines: " + lnr.getLineNumber());
}

```

29) What is piped I/O used for?

The piped I/O streams are for inter-thread communication. They manage the synchronization across thread boundaries of the buffer.

30) How can I show a progress bar while reading a file?

Here's a simple example of monitoring file read progress

```

import java.awt.FlowLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.FileInputStream;
import java.io.InputStream;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.ProgressMonitorInputStream;

public class Test {
    public static void main(String[] args) {
        // create a test frame with a "press me" button
        final JFrame f = new JFrame("Sample");
        f.getContentPane().setLayout(new FlowLayout());
        JButton b = new JButton("Press me");
        f.getContentPane().add(b);
        f.pack();

        // set up the file read action
        b.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                // when button is pressed, start a new thread
                // to read the file. A new thread is needed because we
                // need to free the GUI update thread to paint the
                // progress monitor
                new Thread() {
                    public void run() {
                        try {
                            // open the file, wrapping it in a ProgressMonitorInputStream
                            InputStream in = new FileInputStream("c:\\bigfile.bin");

```

```

        ProgressMonitorInputStream pm =
            new ProgressMonitorInputStream(f,"Reading the big file",in);
        // read the file. If it's taking too long, the progress
        // monitor will appear. The amount of time is roughly
        // 1/100th of the estimated read time (based on how long
        // it took to read the first 1/100th of the file.)
        // Note that by default, the dialog won't appear unless
        // the overall estimate is over 2 seconds.
        int c;
        while((c=pm.read()) != -1) {
            // do something
        }
        pm.close(); // needs better error handling, of course...
    }
    catch(Exception ex) {
        ex.printStackTrace();
    }
}
}.start ();
});

// display the frame
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
f.setVisible(true);
}
}

```

31) How do I initialize a JTextArea from an input file?

The JTextComponent provides a read() method to initialize the text component:

```

JTextArea ta = new JTextArea();
Reader reader = new FileReader(filename);
ta.read(reader, null);

```

The second parameter is a description.

32) How to make a file read only?

For any type of file, just use the setReadOnly() method of File to mark a file as read-only. There is no argument to make it writable once it is read-only.

## 1) What is Unicode?

Unicode provides a standard encoding for the character sets of different languages. The encoding is independent of the platform, program, or language used to access said data. Visit the Unicode Consortium (<http://www.unicode.org>) to learn more about Unicode developments.

## 2) How can I get the number of days that have elapsed between two Date objects?

That depends on what you mean by "between".

If you want to find out the number of 24-hour periods between two Date objects d1 and d2 (d2 > d1) then you would do this:

```
double days = (d2.getTime()-d1.getTime())/1000/60/60/24;
```

Since Date.getTime returns milliseconds, you divide by 1000 to get seconds, by 60 to get minutes, by 60 again to get hours, and by 24 to get days.

Sometimes this can cause difficulties, especially in countries that have a form of daylight savings time. For example, in the U.S., there is one day in the Fall which has 25 hours, and one day in the Spring which has 23 hours. An elapsed-number-of-days calculation on Dates falling on one of these days may not give the answer you expect.

Anyway, that's the easy way, and if you're satisfied with that, then read no further.

If, on the other hand, you mean the number of midnight-crossings (so that the number of days elapsed between 11:00 PM and 1:00 AM the next day is 1) then you're better off using the Calendar class and computing the Julian Day.

By the way, don't mistake the Julian Day for the Julian Calendar. The two are different and named after different "Julians".

The Julian Day is defined as the number of days elapsed since Nov 24, 4714 BC, 12:00 GMT Gregorian. The year was chosen as being sufficiently in the past so as not to have negative Julian Days, the date was chosen because of the (in that year) 37-day difference between the Julian and Gregorian calendars (Nov 24, 4714 BC Gregorian would be Jan 1, 4713 BC Julian), and the time was chosen because astronomers do their work at night, and it would be a little confusing (to them) to have the Julian Day change in the middle of their work. (I could be wrong)

Anyway, the algorithm for computing the Julian Day from a Gregorian or Julian calendar date for years 400 AD and above is as follows:

Let the date be Y, M, D, where Y is the AD year, M is the month (January=1, December=12), and D is the day (1-31).

For the following calculations, use integer arithmetic (i.e. lop off the fractional part of any result).

If  $M==1$  or  $M==2$  then  $Y--$ ,  $M+=12$ .  
Let  $A = Y/100$   
Let  $B = A/4$   
Let  $C = 2-A-B$  (or  $C=0$  if you're using the Julian calendar)  
Let  $E = 365.25*(Y+4716)$   
Let  $F = 30.6001*(M+1)$

Julian Day =  $C+D+E+F-1524.5$

There would be a further adjustment for the time of day, but we're not looking at that. If you want your Julian Days to start at midnight, subtract 1524 rather than 1524.5.

And so, your number-of-days-elapsed calculation would look like this:

```
Calendar c1 = new GregorianCalendar ();  
Calendar c2 = new GregorianCalendar();
```

```
c1.setTime (d1);  
c2.setTime (d2);
```

```
long j1 = julian_day(c1.get(YEAR),  
    c1.get(MONTH)+1, c1.get(DAY));  
long j2 = julian_day(c2.get(YEAR),  
    c2.get(MONTH)+1, c2.get(DAY));
```

```
long days_elapsed = j2-j1;
```

And this would properly calculate the number of midnight-crossings between 11:00 PM on one day and 1:00 AM the next day.

You can get a little more information about the Julian Day calculations [here](#).

You can get much more information about different calendar systems [here](#), [here](#), and [here](#).

3) What is a ResourceBundle and what it is used for?

A ResourceBundle is like a specialized form of a Hashtable that maps strings to values. That doesn't sound too exciting by itself, but the magic of ResourceBundle allows you to have different lookup tables based upon what Locale (language/country) a user is coming in from. And, if a setting doesn't change between languages/countries, there is no need to repeat the setting in the customized bundle, as the runtime will search through several bundles until it finds a mapping (though in theory it could find none). Resource bundles also support another level of variants beyond language/country, such that you can have customized labels/messages for people in California and New York.

4) How can I get yesterday's date?



```
// get Calendar with current date
java.util.GregorianCalendar gCal =
    new GregorianCalendar();

// get yesterday's date
gCal.add( Calendar.DATE, -1 );

// get components of yesterday's date
String sMonth = ( gCal.get( Calendar.MONTH ) + 1 ) + "";
String sDay  = gCal.get( Calendar.DATE ) + "";
String sYear = gCal.get( Calendar.YEAR ) + "";

// get yesterday's date in milliseconds
long lMillis = gCal.getTime().getTime();
```

#### 5) How do I display and parse a date?

The Java 1.8N way is to use a `DateFormat`. While `SimpleDateFormat`, which is generally returned, creates a large number of objects, it is locale aware and will handle most of your needs. The following sample code initially creates a `java.sql.Date` object and formats it for the default locale. An initial `ActionPerformed` call additionally formats/displays it for a German locale and also displays the resulting `java.sql.Date` in standard escape format. Other dates can be entered and parsed after the initial display.

// JDFDP.java - Display and Parse `java.sql.Date`

```
import java.sql.*;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.text.*;
import java.util.*;

public class JDFDP extends JFrame
    implements ActionListener,
        WindowListener
{
    // create a java.sql.Date
    java.sql.Date jsqDate = new java.sql.Date(
        System.currentTimeMillis() );

    DateFormat dfLocal = DateFormat.getDateInstance(
        DateFormat.SHORT );
    DateFormat dfGermany = DateFormat.getDateInstance(
        DateFormat.SHORT, Locale.GERMANY );

    JButton    jb = new JButton( "Go" );
    JLabel     jll = new JLabel("Input a Date:");
    JLabel     jID = new JLabel("Display German:");
    JLabel     jIP = new JLabel("Parsed:");
```

```

JPanel    jp = new JPanel();

JTextField jtl = new JTextField( 10 ),
           jtD = new JTextField( 10 ),
           jtP = new JTextField( 10 );

public JDFDP()
{
    super( "JDFDP" );
    addWindowListener( this );

    jb.addActionListener( this );

    jp.add(jl);
    jp.add(jtl);
    jp.add(jb);
    jp.add(jlD);
    jp.add(jtD);
    jp.add(jlP);
    jp.add(jtP);

    getContentPane().add( jp, BorderLayout.CENTER );
    pack();

    // set text by sending dummy event
    jtl.setText( dfLocal.format( jsqldate ) );
    actionPerformed(
        new ActionEvent( this, 12, "12" ) );

    show();
} // end constructor


// ActionListener Implementation
public void actionPerformed(ActionEvent e)
{
    jtD.setText( "" );
    jtP.setText( "" );
    try
    {
        java.util.Date d = dfLocal.parse(
            jtl.getText() );
        jtl.setText( dfLocal.format( d ) );
        jtD.setText( dfGermany.format( d ) );
        d = dfGermany.parse( jtD.getText() );
        // get new java.sql.Date

```

```

        jsqDate = new java.sql.Date( d.getTime() );

        jtP.setText( jsqDate.toString() );
    }
    catch( ParseException pe ) { jtl.setText( "" ); }

} // End actionPerformed

// Window Listener Implementation
public void windowOpened(WindowEvent e) {}
public void windowClosing(WindowEvent e)
{
    dispose();
    System.exit(0);
}
public void windowClosed(WindowEvent e) {}
public void windowIconified(WindowEvent e) {}
public void windowDeiconified(WindowEvent e) {}
public void windowActivated(WindowEvent e) {}
public void windowDeactivated(WindowEvent e) {}
// End Window Listener Implementation

public static void main(String[] args)
{
    new JDFDP();
}

} // end class JDFDP

```

## Serialization

### 1) What is object serialization?

*Serializing* an object involves encoding its state in a structured way within a byte array. Once an object is serialized, the byte array can be manipulated in various ways; it can be written to a file, sent over a network using a socket-based connection or RMI, or persisted within a database as a BLOB.

The serialization process encodes enough information about the object type within the byte stream, allowing the original object to be easily recreated upon *deserialization*, at a later point in time.

### 2) Can I persist my objects using serialization instead of using a relational or object database?

No. While serialization is a highly versatile mechanism having numerous applications, your long term storage needs should continue to be addressed by conventional relational or object databases.

Note that serialization does not provide any features for transaction management and concurrency control. Nor does it provide typical database features like indexed access, caching and a query language.

3) Why doesn't serialization save the value of static variables?

Variables declared as static members are not considered part of the state of an object because they are shared by all instances of that class. Classes which need to preserve the value of static members during serialization should save and restore these values explicitly using private void `readObject(ObjectInputStream)` and private void `writeObject(ObjectOutputStream)`.

4) What are the advantages and disadvantages of serialization?

The advantages of serialization are:

- It is easy to use and can be customized.
- The serialized stream can be encrypted, authenticated and compressed, supporting the needs of secure Java computing.
- Serialized classes can support coherent versioning and are flexible enough to allow gradual evolution of your application's object schema.
- Serialization can also be used as a mechanism for exchanging objects between Java and C++ libraries, using third party vendor libraries (like RogueWave's Tools.h++ ) within C++.
- There are simply too many critical technologies that rely upon serialization, including RMI, JavaBeans and EJB.

However, serialization has some disadvantages too:

- It should ideally not be used with large-sized objects, as it offers significant overhead. Large objects also significantly increase the memory requirements of your application since the object input/output streams cache live references to all objects written to or read from the stream until the stream is closed or reset. Consequently, the garbage collection of these objects can be inordinately delayed.
- The `Serializable` interface does not offer fine-grained control over object access - although you can somewhat circumvent this issue by implementing the complex `Externalizable` interface, instead.
- Since serialization does not offer any transaction control mechanisms per se, it is not suitable for use within applications needing concurrent access without making use of additional APIs.

5) What things are required for a class that implements `Serializable`?

A class that implements Serializable must also:

- Have access to a no-argument constructor in its first non-serializable superclass
- Identify non-serializable data members using the transient keyword or explicitly mark data members as serializable using the serialPersistentFields member.

- 6) My subclass implements Serializable but my superclass doesn't. Both subclass and superclass contain instance variables that need to be saved as part of the state of the subclass. Will serialization save the superclass fields for me?

When you serialize an object, the serialization mechanism works by chaining up the inheritance hierarchy, saving the state of each Serializable superclass in turn. When serialization reaches the first non-serializable superclass, the serialization stops.

When deserializing, the state of this first non-serializable superclass is restored not from the stream, but by invoking that class' no-argument constructor. If the no-argument constructor is not adequate for your purposes, you must customize the serialization of your subclass with writeObject() and readObject() in order to write out and restore any information from the non-serializable superclass that you find necessary.

- 7) What role does serialization have in RMI?

RMI uses serialization as its basic and *only* mechanism for sending objects across a network.

If an object implements java.rmi.Remote, then the object's stub is serialized and sent to the client. If the object implements java.io.Serializable, then the object itself is serialized and sent.

- 8) Why would I want to implement Externalizable instead of Serializable?

By implementing Externalizable yourself you can win performance at the cost of flexibility and extra code to maintain. If you implement Externalizable yourself you stream the data directly without the need for reflection which is used in the case of Serializable.

- 9) How can I read and write serialized objects to and from a database?

If your RDBMS supports them, you can store serialized objects as BLOBs.

These are JDBC 2.0 features, but take a look at java.sql.Blob, ResultSet and PreparedStatement for more information.

10) Can a Vector or a Hashtable be serialized and deserialized?

Both these classes implement Serializable, and have been designed for serialization. So yes, they can be serialized.

One thing you have to watch out for, though, is that in order to serialize a collection like Vector or Hashtable, you must also be able to serialize all of the objects *contained* in these collections. Otherwise, the collection would not be able to be completely restored. Your program will throw a NotSerializableException unless *all* objects stored in the Vector or Hashtable are also serializable.

11) Is there any way to save the state of an object of a class which does not implement Serializable or Externalizable?

Yes. You will have to write the value of each and every instance variable into the persistent storage. If there are 100 variables, you will have to store each of them individually. If some of the variables are object references, you will have to follow each reference and save the state of that object as well. You can do that, but it would be a proprietary solution and each class that wanted to *read* your object would also have to know all about your proprietary format.

12) What is the role of serialization in EJB?

A big part of EJB is that it is a framework for underlying RMI: remote method invocation. You're invoking methods remotely from JVM space 'A' on objects which are in JVM space 'B' -- possibly running on another machine on the network.

To make this happen, all arguments of each method call must have their current state plucked out of JVM 'A' memory, flattened into a byte stream which can be sent over a TCP/IP network connection, and then deserialized for reincarnation on the other end in JVM 'B' where the actual method call takes place.

If the method has a return value, it is serialized up for streaming back to JVM A. Thus the requirement that all EJB methods arguments and return values must be serializable. The easiest way to do this is to make sure all your classes implement `java.io.Serializable`.

## Security

1) What is "private key"/"symmetric"/"secret key" cryptography?

A private key cryptography algorithm uses the same key for encryption and decryption. Since the key is the only data required to decrypt the ciphertext, it must be kept private/secret.

Encryption; creating ciphertext from plaintext with a specific encryption algorithm and an encryption key

Decryption; recovering plaintext from ciphertext using a specific encryption algorithm and a decryption key

## 2) What is "public key"/"asymmetric" cryptography?

A public key cryptography algorithm uses two different (but related) keys for encryption and decryption. The key used for decryption is kept secret (Private) whereas the encryption key can be distributed openly (Public). Thus, anyone in possession of the public encryption key may encrypt and send a message to the holder of the private decryption key. However, only the holder of the private decryption key may decipher the message. Both keys must be created and used in conjunction, and are often referred to as a key pair.

## 3) What is hybrid cryptography?

Combination of public and private key cryptography, where the exchange of an encrypted session key is done using public key cryptography. The following encrypted session is then pursued with private/symmetric key cryptography. The main reason is that private key cryptography is generally much quicker than public key cryptography.

## 4) What is a Message Digest?

A Message Digest is a digitally created hash (fingerprint) created from a plaintext block. All the information of the message is used to construct the Message Digest hash, but the message cannot be recovered from the hash. For this reason, Message Digests are also known as *one way hash functions*.

The size of a Message Digest is always the same, independent of the size or content of the message from which it was created. Generally, the size of a Message Digest is fairly short (1024 bits). The ideal Message Digest algorithm would possibly alter 50% of the bits in the resulting hash if one bit was altered in the plaintext message.

## 5) What is a Digital Signature?

Digital Signatures are used to ensure the identity of a sender. In conjunction with Message Digests, Digital Signatures prevents someone from altering a message and falsely claiming you wrote the altered message. Digital Signatures are a byproduct of public key cryptography, as demonstrated below. (Believe me, it is simpler to describe the concept of a Digital Signature by taking an example).

## 6) What do you need to establish computer security?

There are essentially four aspects of computer security:

Aspect	Generally achieved by	Explanation	Effect
Confidentiality	Encryption	Protects message content from any but sender and recipient	Transmission eavesdroppers cannot read or use the message content.
Integrity	Message Digest	Verifies that a message has not been modified since it was sent	Transmission hijackers cannot modify message in transit
Authenticity	Digital Signature + Password	Uniquely identifies the sender of a message	Verification of actual sender possible
Nonrepudiation	Cryptographic Receipt	Prevents a sender from falsely denying sending a message	Message recipient can prove that a message was sent by sender

## 7) What is SSL?

SSL stands for Secure Socket Layer. It is a protocol developed by Netscape for encrypting information sent between processes over TCP/IP sockets. It sits between application software and the TCP/IP sockets. You'll find it frequently used between web browsers and web servers using the *https* URL prefix, providing encryption, integrity, authentication, and non-repudiation.

## 8) How can I support HTTPS (SSL) in a servlet?

The servlet technology by design already supports https (SSL). However, the way this works is not through the servlet technology but through the Web Server. The web server controls whether information is done securely (https) versus non-securely (http).

One way to force servlets to go down the https path is to define your web server to only allow secure connections when accessing servlets. In IIS this can be accomplished through the definition of ISAPI filters. The ISAPI filter can instruct the web server to route all requests that end with a pre-defined prefix to the servlet engine. The trick is to then define files, with the predefined extension, in the web servers directory. For example, if the servlet's name is MyServlet a file with the name MyServlet.xxx would be placed on the web server. All calls to this file would be routed to the servlet engine. And IIS would be used to force all calls



to the MyServlet.xxx file to go through https. The JRun servlet engine has examples of how to do this documented on their web page.

## Applets

- 1) How can I download a file (for instance, a Microsoft Word document) from a server using a servlet and an applet?

Try telling the applet to call  
`getAppletContext().showDocument("http://foo.com/whatever/blah.doc")`. That makes the browser responsible for fetching and saving the document (including putting up the "Save As" dialog).

You may also want to check the web server configuration to make sure the appropriate MIME-type configuration is set. If your servlet is the one fetching the document (or creating it on the fly), then it must set the MIME-type using `response.setContentType("application/ms-word")` or equivalent.

- 2) How do I get an applet to load a new page into the browser?

In short, get the applet's context and tell it to show a different document:  
`getAppletContext().showDocument(new URL("http://foo.com/whatever/blah.doc"))`. You can also provide a target frame string as in `getAppletContext().showDocument(new URL("http://foo.com/whatever/blah.doc"), "target")`. There are four special targets strings. Any other string will act as a named top-level window, if the frame doesn't exist.

"_self"	Show in same frame as applet
"_parent"	Show in the applet's parent frame
"_top"	Show in the top-level frame of the applet's window
"_blank"	Show in a new, unnamed top-level window

- 3) What is the syntax of the <APPLET> tag?

<APPLET

// Required

CODE = *appletClass* or

OBJECT = *serializedApplet*

HEIGHT = *pixels*

WIDTH = *pixels*

// Options

CODEBASE = *codebaseURL*

ARCHIVE = *archiveList*

ALT = *alternateText*

NAME = *instanceName*

ALIGN = *alignment*

```

HSPACE = pixels
VSPACE = pixels
>
<PARAM NAME = attribute1 VALUE = value1>
<PARAM NAME = attribute2 VALUE = value2>
...
Alternate HTML
</APPLET>

```

Support for OBJECT is limited.

#### 4) How do I create an applet that accepts an unlimited number of parameters?

While there is some upper bounds to the number of parameters the browser will let you specify, if you would like to let a user specify an undefined number of values, you can let them specify a parameter with a number at the end and keep counting until no more parameters following that pattern are there, as in:

```

<applet code=Params width=10 height=10>
<param name=Param0 value=Hello>
<param name=Param1 value=World>
</applet>

```

To do this, you just increment a counter for the `getParameter()` calls:

```

import java.applet.*;

public class Params extends Applet {
    public void init() {
        String value;
        int i = 0;
        while ((value = getParameter("Param" + i)) != null) {
            System.out.println ("Param" + i + ": " + value);
            i++;
        }
    }
}

```

#### 5) How can I connect from an applet to a database on the server?

There are two ways of connecting to a database on the server side.

1. The hard way. Untrusted applets cannot touch the hard disk of a computer. Thus, your applet cannot use native or other local files (such as JDBC database drivers) on your hard drive. The first alternative solution is to [create a digitally signed applet](#) which may use locally installed JDBC drivers, able to connect directly to the database on the server side.

2. The easy way. Untrusted applets may only open a network connection to the server from which they were downloaded. Thus, you must place a database listener (either the database itself, or a middleware server) on the server node from which the applet was downloaded. The applet would open a socket connection to the middleware server, located on the same computer node as the webserver from which the applet was downloaded. The middleware server is used as a mediator, connecting to and extract data from the database.
- 6) How do I display a message in the browser's status bar?

To show a message in the status bar, you would call the `showStatus()` method of the `AppletContext`:

```
getAppletContext().showStatus(message);
```

Keep in mind that the browser can overwrite the message at any time or the browser might not have a visible status bar. This basically means you should not use the status bar for messages that might be missed.

- 7) What are the main differences between applets and applications?

Applets are created to be embedded within a browser. As such, they have methods that are automatically called by the browser at certain times: `init()` when first loaded, `start()` when page loaded/entered, `stop()` when page left, `destroy()` when applet unloaded, and `paint()` when the screen needs to be displayed. In addition, applets have severe security restrictions as they can be loaded from untrusted sources over the Internet. Applications on the other hand need to be installed locally, start execution with the `main()` method, and have no inherent security restrictions. Beyond this, applications control their own operations.

- 8) How can I get the real local host IP address in an applet?

Applet security restrictions do not let you get this in an untrusted applet via `InetAddress.getLocalHost()`.

However, you can get this address by creating a `Socket` connection back to the web server from which you came and asking the `Socket` for the local address:

```
URL url = getDocumentBase();
String host = url.getHost();
Socket socket = new Socket(host, 80);
InetAddress addr = socket.getLocalAddress();
String hostAddr = addr.getHostAddress();
System.out.println("Addr: " + hostAddr);
```

- 9) How do I load a serialized applet?

Instead of specifying the CODE attribute of an <APPLET> tag, you use the OBJECT attribute, as in:

```
<APPLET
  OBJECT=TheApplet.ser
  WIDTH=300
  HEIGHT=300
>
</APPLET>
```

This allows you to preinitialize your applet to some saved state, instead of having to reinitialize your applet each time it is loaded.

Support for this in browsers is severely limited. Don't rely on it.

10) From one applet, how do I communicate with another applet loaded from the same server?

It really doesn't matter that they are loaded from the same server, as long as they are running in the same JVM in the browser, you can call methods between them.

If you include a *name="appletA"* value in the applet tag of the applet you want to call methods on, you get a reference to it by using the AppletContext:

```
MyApplet appl = (MyApplet) getAppletContext().getApplet("appletA");
if (null != appl) {
    appl.register("Hello there!", this);
}
```

Checking that you get a reference to the class you expect is left as an exercise to the reader :-)

11) In an applet, is the start() or paint() method executed first?

The initial order in which the lifecycle methods of applets are executed when loaded is:

```
init()
start()
paint()
```

And when done:

```
stop()
destroy()
```

This also holds when you call `repaint()` in the `init()` method. However, the `init()` method will not be called if you load a serialized applet.

12) What are the main differences between JavaBeans and applets?

Applets are Java programs which are meant to be run by an applet-runner (which is typically inside a web browser). By definition, applets are applets because they inherit from `java.applet.Applet`. Also, applets are pretty constrained in what they are/aren't allowed to do.

Traditional JavaBeans are Java software *components*. A bean may be a single, simple, standalone class or it may be a complicated behemoth but, as long as it follows the rules and conventions of JavaBeans, then it's a JavaBean.

13) How do I use an applet that resides on a remote machine?

You would specify the location of the class files in the `CODEBASE` attribute of the `<APPLET>` tag:

```
<applet code=MyApplet
width = 300 height = 300
CODEBASE="http://www.jguru.com/appletdirectory/">
</applet>
```

14) When you communicate with a servlet from an Applet or an application, how can you ensure that the session information is preserved? That is, how do you manage cookies in applet-servlet communication?

For sessions, your best bet is to have the servlet rewrite the URLs to include the session information. With cookies read [I am writing an application that retrieves a URL, but that URL requires a cookie. Is there a way to hold that cookie within my application and send it back to the URL when requested?](#) Essentially, you can send the cookie back easily with HTTP. For instance, `urlConnection.setRequestProperty("cookie", _cookie);` Where the `_cookie` is coming from a previous call: `String _cookie = urlConnection.getHeaderField("set-cookie");`

15) How can I change my application into applet?

- Create a subclass of `java.applet.Applet`
- Take the code in the `main()` method of the application and move it to the `init()` method of the applet
- Create an HTML file to load the applet
- If necessary, move any requirements that untrusted applets cannot do into an appropriate server-side task
- If necessary, modify all getting of support files
- Test

16) In detail, how do the applet life cycle methods `init()`, `start()`, `paint()`, `stop()`, and `destroy()` work?

The life cycle methods of an applet are `init()`, `start()`, `stop()`, and `destroy()`. While frequently grouped with the methods, the `paint()` method is not really part of the applet's life cycle. The method is just called by the browser whenever a part of the screen has become invalidated or in response to a programmer-generated repaint request. The `update()` method falls into the same category as `paint()`.

As far as the other methods go:

- `init()` - called once when applet is loaded
- `start()` - called every time the page the applet is located on is loaded, after `init()` or after leaving the page and returning
- `stop()` - called every time the page the applet is located on is left
- `destroy()` - called when applet is unloaded

If loading a serialized version of an applet from a `.ser` file (OBJECT attribute in `<APPLET>` tag instead of CODE attribute), the `init()` method is not called.

Use the `start()` / `stop()` method pair to deal with starting and stopping threads, as when the page isn't active it isn't nice to use up resources.

Don't rely on the `destroy()` method being called to free a crucial resource, in case the browser crashes.

I had heard that `start()` / `stop()` were called for iconifying and deiconifying, but a test with the latest browser versions shows they aren't called.

17) How does an applet get loaded into the browser?

The browser's HTML interpreter reads the parameters from the `APPLET` tag. The relevant parameters used to load the applet's class are `CODE`, `CODEBASE`, and `ARCHIVE` (`CODEBASE` and `ARCHIVE` are optional - `OBJECT` can be used instead of `CODE`). The `CODE` parameter holds the name of the applet's class file. The `CODEBASE` holds the class file directory (the default is the current directory), and the `ARCHIVE` is the name of a jar or zip file that the applet's class files are stored in (the default is no archive).

The browser then reads the byte code from the class files and performs a verification pass, checking for access violations and invalid opcodes (among other things). The Java Virtual Machine (JVM) then builds an object for each class (provided the objects have not been created by a previous instance), and executes any static code in each object's initialization section. Object initialization also includes the construction of the superclass objects (the superclass files from `Applet` to `Object` are typically loaded from the browser's own Java API on the local file system).

If the applet makes it this far without generating security exceptions, missing class errors, or common errors in static code segments, the JVM will make calls to initialization functions in the superclass, then to the applet's `init()` member.

18) What is needed in the `APPLET` tag to cause an applet to be loaded from a JAR file?

To specify that an applet loads from a JAR file, instead of from separate classes, add an `ARCHIVE` attribute to your `<APPLET>` tag:

```
<APPLET
  CODE=AppletClass
  ARCHIVE="foo.jar"
  WIDTH=200
  HEIGHT=200>
</APPLET>
```

If you need to specify multiple JAR files, provide a comma separated list.

19) When should one use applets and when should one use servlets?

Applets run inside browsers, and servlets run inside servers. So broadly speaking, use applets when you need dynamism on the client side, and use servlets when you need dynamism on the server side.

Servlets can produce any HTML (or indeed, any file type), and therefore are much more versatile than applets. Servlets can tailor the HTML they produce based on the browser type (as specified in the "User-Agent" header), and thus can produce output that will work in virtually any client, including WAP browsers or the like. Applets are notoriously poorly supported, and even in browsers that ostensibly have Java support, don't work consistently.

Servlets can also produce HTML with embedded JavaScript, which allows a fair amount of dynamic behavior on browsers which support it.

However, due to the infuriating limitations and bugs in JavaScript, sometimes you've just got to have an applet to get the client-side behavior you want. In these cases, take comfort in the fact that you can generate HTML containing `APPLET` tags from inside a servlet.

20) Why do I not create a constructor for an applet?

One doesn't see constructors in applets because the applet container doesn't guarantee a complete environment (valid applet context) until the `init()` method is called.

21) Why doesn't `appletviewer appletclass.class` work?

Applets are loaded with an HTML file. You need to specify an <APPLET> tag in the .html file to load the APPLET.

22)What are the differences between trusted and untrusted applets?

In general, applets loaded over the net are prevented from accessing files on the client machine, and from making network connections except to the host originating the applet.

In addition, applets loaded over the network are prevented from starting other programs on the client. Applets loaded over the network are also not allowed to load libraries, or to define native method calls, providing accessing to the underlying computer.

On the other hand, trusted applets can be permitted to do all that. The trust model in Java allows you to enable only certain operations, so it is possible that you enable network connections to/from anywhere but don't permit access to the local file system.

How to enable trusting an applet is different for each browser, IE, Netscape, and standard Java with the Plug-in.

23)Why doesn't an applet viewed through the browser respond to keyboard events, but the same works fine from within appletviewer?

In order to respond to keyboard events your applet needs to have the input focus. In the case of appletviewer, there is nothing else on the page that might have focus. In the case of a browser, there is. Just click into the applet's area to have it get the input focus and then it will respond to keyboard events.

24)What is the Java Plug-In?

The Plug-in is a replacement Java Runtime Environment for your browser. It allows you to upgrade the browser's supported Java version without waiting for the browser vendor to release a new version of the browser.

25)How do I automatically set the input focus in my applet to a specific component/text field when the browser loads the page that contains the applet?

The java.awt.Component class has a method named requestFocus(), you can use this to request focus on any component.

26)What is an applet stub?

An applet stub is an interface that the applet container (browser) creator will implement abstracting out capabilities like getting applet parameters, the applet's



code/document base and other information necessary for the contained applet(s). As an applet program, you'll rarely use the interface. The only time you really run across a need to do anything with the interface is if you try to run your applet in your own container. You'll then have to implement the AppletStub interface yourself, along with the AppletContext interface.

27)How do I stop an applet from running?

Browsers will constantly call the applet life-cycle methods. There is no way to stop them. The only way to stop an applet from running / displaying anything / taking up CPU time is to set a flag in the code and check it within each of the life-cycle methods. Calling destroy() is not something you as a user / developer should do. Calling the method is the responsibility of the applet container (the browser) when it wants to unload the applet.

28)How can an applet get information about the URL it was called from?

The getDocumentBase() method of Applet returns the base URL of the HTML file that loaded an applet. It is frequently used in conjunction with getting supplemental files like images, as in `Image image = getImage(getDocumentBase(), "foo.gif");`

While in most cases it returns the full URL of the HTML file that loaded the applet, the API docs specify that it only needs to return the URL for the directory of the file.

29)What is an applet?

Basically, an applet is a Java class, extending from `java.applet.Applet` that runs in a Web browser, loaded from an `APPLET` tag in an HTML file. The browser then calls applet life-cycle methods like `start()`, `stop()`, `init()`, and `destroy()` as needed. There are security restrictions untrusted applets must abide by, like not being able to connect to anywhere besides the web server from which it came.

30)What API changes are there for applets with JDK 1.4?

The constructor for the Applet class now throws a `HeadlessException`.

Also, the AppletContext interface now has three new methods: `setStream()`, `getStream()`, and `getStreamKeys()`.

31)Are there any restrictions placed on applets loaded from HTML pages generated from servlets / JSP pages?

It is good practice to include the `CODEBASE` attribute in the `APPLET` tag and specify a directory other than the servlet directory for storing the applet-related classes. For example:

```
out.println(<"APPLET CODE=Foo.class CODEBASE=/applets WIDTH=400  
HEIGHT=400">);
```

Some Web servers think that all classes loaded from the servlets directory are automatically servlets, thus causing the applet not to load.

32)How can I easily provide default values for when applet parameters are not specified?

Here's a helper method that you can use to do that:

```
import java.applet.*;  
  
public class AppletUtils {  
    private AppletUtils() {}  
  
    public static String getParameter(  
        Applet applet, String key, String keyDefault) {  
  
        String value = applet.getParameter(key);  
  
        return (value == null) ? keyDefault : value;  
  
    }  
}
```

Here's how to call:

```
String var = AppletUtils.getParameter(this, "name", "Jaeger");
```

33)What is the default layout manager of an applet?

For a java.applet.Applet, the default layout manager is FlowLayout.

For the content pane of a javax.swing.JApplet, the default layout manager is a BorderLayout.

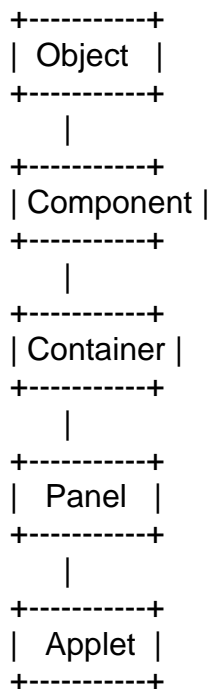
34)My applet runs fine with appletviewer but not in the browser, what can be the problem?

Applets loaded through appletviewer are more then likely loading their classes from the local CLASSPATH, not over the network. As such, they get different permissions and can do things that applets loaded over the network cannot. Try using Tomcat and serving the Applet through an HTTP connection, instead of loading the .HTML file from the file system.

Also, Java runtimes in browsers are based on Java 1.1.x. If you need to use newer capabilities, you'll need to use the Java Plug-in. Without it, you're stuck with the older Java runtime capabilities.

35)What does the class hierarchy of the Applet class look like?

The inheritance hierarchy of the java.applet.Applet class is as follows:



36)How can I maintain a single instance of an object in an applet?

In start(), instead of always creating a new object, return the existing one if it exists or create a new one if it doesn't.

### Networking

1)What is UDP and how does it work?

UDP stands for User Datagram Protocol. UDP provides an unreliable packet delivery system built on top of the IP protocol. As with IP, each packet is an individual, and is handled separately. Because of this, the amount of data that can be sent in a UDP packet is limited to the amount that can be contained in a single IP packet. Thus, a UDP packet can contain at most 65507 bytes (this is the 65535-byte IP packet size minus the minimum IP header of 20 bytes and minus the 8-byte UDP header).

UDP packets can arrive out of order or not at all. No packet has any knowledge of the preceding or following packet. The recipient does not acknowledge

packets, so the sender does not know that the transmission was successful. UDP has no provisions for flow control--packets can be received faster than they can be used. We call this type of communication *connectionless* because the packets have no relationship to each other and because there is no state maintained.

The destination IP address and port number are encapsulated in each UDP packet. These two numbers together uniquely identify the recipient and are used by the underlying operating system to deliver the packet to a specific process (application). Each UDP packet also contains the sender's IP address and port number.

One way to think of UDP is by analogy to communications via a letter. You write the letter (this is the data you are sending); put the letter inside an envelope (the UDP packet); address the envelope (using an IP address and a port number); put your return address on the envelope (your local IP address and port number); and then you send the letter.

Like a real letter, you have no way of knowing whether a UDP packet was received. If you send a second letter one day after the first, the second one may be received *before* the first. Or, the second one may never be received.

## 2) What is a Datagram?

Datagram is another name for a UDP packet.

Java provides two classes for explicitly dealing with datagrams, `DatagramSocket` and `DatagramPacket`. These are both found in the `java.net` package.

## 3) Why use UDP if it is unreliable?

Two reasons: speed and overhead. UDP packets have almost no overhead--you simply send them then forget about them. And they are fast, because there is no acknowledgement required for each packet. Keep in mind that unreliable doesn't mean that packets can be lost or misdirected for no reason - it simply means that UDP provides no built-in checking and correction mechanism to gracefully deal with losses caused by network congestion or failure.

UDP is appropriate for the many network services that do not require guaranteed delivery. An example of this is a network time service. Consider a time daemon that issues a UDP packet every second so computers on the LAN can synchronize their clocks. If a packet is lost, it's no big deal--the next one will be by in another second and will contain all necessary information to accomplish the task.

Another common use of UDP is in networked, multi-user games, where a player's position is sent periodically. Again, if one position update is lost, the next one will contain all the required information.

A broad class of applications is built on top of UDP using streaming protocols. With streaming protocols, receiving data in real-time is far more important than guaranteeing delivery. Examples of real-time streaming protocols are RealAudio and RealVideo which respectively deliver real-time streaming audio and video over the Internet. The reason a streaming protocol is desired in these cases is because if an audio or video packet is lost, it is much better for the client to see this as noise or "drop-out" in the sound or picture rather than having a long pause while the client software stops the playback, requests the missing data from the server. That would result in a very choppy, bursty playback which most people find unacceptable, and which would place a heavy demand on the server.

#### 4) What is Multicast and how does it work?

TCP and UDP are both *unicast* protocols; there is one sender and one receiver. Multicast packets are a special type of UDP packets. But while UDP packets have only one destination and only one receiver, multicast packets can have an *arbitrary* number of receivers.

Multicast is quite distinct from broadcast; with broadcast packets, *every* host on the network receives the packet. With multicast, only those hosts that have registered an interest in receiving the packet get it.

This is similar to the way an AWTEvent and its listeners behave in the Abstract Window Toolkit. In the same way that an AWTEvent is sent only to registered listeners, a multicast packet is sent only to members of the *multicast group*. AWT events, however, are unicast, and must be sent individually to each listener--if there are two listeners, two events are sent. With a MulticastSocket, only one is sent and it is received by many.

As you might guess, MulticastSocket is a subclass of DatagramSocket which has the extended ability to join and leave multicast groups. A multicast group consists of both a multicast address and a port number. The only difference between UDP and multicast in this respect is that multicast groups are represented by special internet addresses in the range 224.0.0.1 to 239.255.255.255, inclusive. Just as there are well-known ports for network services, there are reserved, well-known multicast groups for multicast network services.

When an application subscribes to a multicast group (host/port), it receives datagrams sent by other hosts to that group, as do all other members of the group. Multiple applications may subscribe to a multicast group and port concurrently, and they will all receive group datagrams.

When an application sends a message to a multicast group, all subscribing recipients to that host and port receive the message (within the time-to-live range of the packet, see below). The application needn't be a member of the multicast group to send messages to it.

#### 5) What are RFCs and where can I find them?

RFC stands for "Request for Comment". The RFCs form an integral part of the Internet standards; standards are formed by first publishing a specification as an RFC.

If you wish to implement a standard protocol in Java, the first thing to do is download and read the applicable RFC.

#### 6) How can I send objects across the network using sockets?

Objects that implement `Serializable` may be sent across a socket connection using an `ObjectInputStream` and `ObjectOutputStream` combination.

Here are the steps to follow:

1. First, define an object to send. As an example, we can define a class called `Message` to encapsulate our communications:
2.     `public class Message implements Serializable {`
3.         `private int senderID;`
4.         `private String messageText;`
- 5.
6.         `public Message(int id, String text) {`
7.             `senderID = id;`
8.             `messageText = text;`
9.         `}`
10.         `public String getText() {`
11.             `return messageText;`
12.         `}`
13.     `}`
14. Next, instantiate the object, wrap the socket's streams in object streams, then send the message across the socket:
15.     `Message sayhey = new Message("123456789", "Hello");`
- 16.
17.     `Socket socket = new Socket(host, port);`
18.     `ObjectOutputStream out = new`
19.         `ObjectOutputStream(socket.getOutputStream());`
20.     `out.writeObject(sayhey);`
21. On the other side of the socket, the message can be retrieved and used by invoking methods on the returned object:
22.     `ObjectInputStream in = new`
23.         `ObjectInputStream(socket.getInputStream());`
24.     `Message messageObject = (Message) in.readObject();`
25.     `String messageText = messageObject.getText();`

#### 7) How can I obtain a hostname given an IP address?

The code snippet:

```
String host = InetAddress.getByName("216.217.9.172").getHostName();
```

should give you the hostname www.jguru.com

#### 8)How can I obtain the IP address for a given hostname?

The code snippet:

```
ip =InetAddress.getByName("www.jguru.com").getHostAddress();
```

should give you the IP address 216.217.9.172

#### 9)How can I download files from a URL using HTTP?

One way to do this is by using a URLConnection to open a stream to your desired URL, then copy the data out of the stream to a file on your local file system. As an example, here's a little snippet of code which lets you download the jGuru logo to your machine:

```
URL url = new URL("http://www.jguru.com/images/logo.gif");
URLConnection connection = url.openConnection();
InputStream stream = connection.getInputStream();
BufferedInputStream in = new BufferedInputStream(stream);
FileOutputStream file = new FileOutputStream("logo.gif");
BufferedOutputStream out = new BufferedOutputStream(file);
int i;
while ((i = in.read()) != -1) {
    out.write(i);
}
out.flush();
```

Note the use of "Stream" classes instead of "Reader" classes. This is done because in general a URL may contain binary data, as in this example. Also note that both the input and the output streams are buffered - this greatly speeds the download time and is far more efficient than reading and writing a single byte at a time, especially over the network.

#### 11) What is TCP and how does it work?

Internet Protocol, or IP, provides an unreliable packet delivery system--each packet is an individual, and is handled separately. Packets can arrive out of order or not at all. The recipient does not acknowledge them, so the sender does not know that the transmission was successful. There are no provisions for flow control--packets can be received faster than they can be used. And packet size is limited.

Transmission Control Protocol (TCP) is a network protocol designed to address these problems. TCP uses IP, but adds a layer of control on top. TCP packets

are lost occasionally, just like IP packets. The difference is that the TCP protocol takes care of requesting retransmits to ensure that all packets reach their destination, and tracks packet sequence numbers to be sure that they are delivered in the correct order. While IP packets are independent, with TCP we can use streams along with the standard Java file I/O mechanism.

Think of TCP as establishing a *connection* between the two endpoints. Negotiation is performed to establish a "socket", and the socket remains open throughout the duration of the communications. The recipient acknowledges each packet, and packet retransmissions are performed by the protocol if packets are missed or arrive out of order. In this way TCP can allow an application to send as much data as it desires and not be subject to the IP packet size limit. TCP is responsible for breaking the data into packets, buffering the data, resending lost or out of order packets, acknowledging receipt, and controlling rate of data flow by telling the sender to speed up or slow down so that the application never receives more than it can handle.

There are four distinct elements that make a TCP connection unique:

- IP address of the server
- IP address of the client
- Port number of the server
- Port number of the client

Each requested client socket is assigned a unique port number while the server port number is always the same. If *any* of these numbers is different, the socket is different. A server can thus listen to one and only one port, and talk to multiple clients at the same time.

So a TCP connection is somewhat like a telephone connection; you need to know not only the phone number (IP address), but because the phone may be shared by many people at that location, you also need the name or extension of the user you want to talk to at the other end (port number). The analogy can be taken a little further. If you don't hear what the other person has said, a simple request ("What?") will prompt the other end to resend or repeat the phrase, and the connection remains open until someone hangs up.

## 12) What's a MalformedURLException?

When you try to create a new URL by calling its constructor, it will throw a MalformedURLException if the URL string is not parseable or contains an unsupported protocol.

## 13) How can I get the real local host IP address in an applet?

Applet security restrictions do not let you get this in an untrusted applet via InetAddress.getLocalHost().



However, you can get this address by creating a Socket connection back to the web server from which you came and asking the Socket for the local address:

```
URL url = getDocumentBase();
String host = url.getHost();
Socket socket = new Socket(host, 80);
InetAddress addr = socket.getLocalAddress();
String hostAddr = addr.getHostAddress();
System.out.println("Addr: " + hostAddr);
```

14) How do I get the real local host IP address in an application (or trusted applet)?

The InetAddress class provides the necessary support:

```
InetAddress localHost = InetAddress.getLocalHost();
System.out.println(localHost.getHostName());
System.out.println(localHost.getHostAddress());
```

15) My machine has more than one IP address. How can I specify which address to use for a socket connection?

Both Socket and ServerSocket have a constructor which allows you to specify a particular local IP address to use. In the case of Socket, there are two constructors:

```
public Socket(InetAddress address, int port, InetAddress localAddr, int localPort)
public Socket(String host, int port, InetAddress localAddr, int localPort)
```

In the case of ServerSocket, the constructor is:

```
public ServerSocket(int port, int backlog, InetAddress localAddr)
```

All three of these constructors may throw an IOException.

16) Are there any performance advantages to be gained by using a Socket to connect to an HTTP server, instead of URL or URLConnection?

Using a Socket gives you more control over the connection - you can set socket options, use your own streams, or even use your own socket implementation factory. However, this comes at great expense; URL and URLConnection do a lot of things for you that you would need to implement from scratch. For example, you would have to completely implement the HTTP protocol, including support for proxies, maintain this code, and keep it up to date as HTTP evolves. URL and URLConnection let you extend functionality through content and protocol

handlers - you would need to implement this mechanism yourself or lose that extensibility. I don't know why you would want to do this - the network overhead for HTTP dwarfs any potential savings in CPU time you might get from dealing with low-level sockets, and you lose generality, flexibility, and maintainability.

If you're convinced you really want control at the socket level, I would suggest that you consider writing and using your own protocol handler instead of reinventing URL and URLConnection.

17) Do I have to explicitly close the socket input and output streams before I close a socket?

While closing the socket will close the underlying socket input and output streams, programs typically wrap these streams in other streams. It is important to flush and close these other streams, preferably in a finally block, before closing the socket. This will help to ensure that all application data is properly sent or received and that your program isn't unnecessarily holding on to file descriptors that it no longer needs. An example of this follows:

```
Socket socket;
BufferedOutputStream out;
try {
    socket = new Socket(host, port);
    out = new BufferedOutputStream(socket.getOutputStream());
    //
    // write to the socket using the "out" stream
    //
}
catch (IOException e) {
    // handle errors
}
finally {
    try {
        out.flush();
        out.close();
        socket.close();
    }
    catch (IOException e) { /*ignore*/ }
}
```

Although the structure of the finally block is a little awkward in this case, it does ensure that the streams and the socket are properly closed down, whether or not an exception occurred.

18) What protocols are supported by the URL and URLConnection classes?

The protocols supported by URL and URLConnection are VM-specific. Sun's JDK supports a different set than the browser VMs or third-party JDK VMs. The protocols that seem to be common to all implementations are HTTP and FTP and file. Other protocols that might be supported include HTTPS, mailto, gopher, and TELNET.

In the table below, some of the more common VMs are listed showing the major protocols they support. There are other protocols not shown in this table that may be supported.

	JDK 1.2	Netscape 4.7	IE 5.0
file	Yes	Yes	Yes
http	Yes	Yes	Yes
https	Yes*	Yes	Yes
ftp	Yes	Yes	Yes
telnet	No	Yes	No
mailto	Yes	Yes	Yes
gopher	Yes	Yes	Yes

\* JDK 1.2 supports HTTPS only when the JSSE extension is installed

If a protocol is not supported in your VM, you can extend the functionality of the URL class by writing and installing your own protocol handler.

#### 19) How can I use the mailto: protocol from URL or URLConnection?

Both URL and URLConnection act as mail user agents when using the mailto: protocol. They need to connect to a mail transfer agent to effect delivery. An MTA, for example sendmail, handles delivery and forwarding of e-mail.

Note that a mailto: URL doesn't contain any information about where this MTA is located - you need to specify it explicitly by setting a Java system property on the command line. For example, to run the program mailto.java shown below, you would use the command:

```
java -Dmail.host=mail.yourisp.net mailto
```

This tells the mailto: protocol handler to open a socket to port 25 of mail.yourisp.net and speak SMTP to it. You should of course substitute the name of your own SMTP server in the above command line.

```
import java.io.*;
import java.net.*;

public class mailto {
```

```
public static void main(String[] args) {
    try {
        URL url =
            new URL("mailto:foo@bar.com");
        URLConnection conn =
            url.openConnection();
        PrintStream out =
            new PrintStream(conn.getOutputStream(), true);

        out.print(
            "From: jguru-fan@yourisp.com"+"\\r\\n");
        out.print(
            "Subject: Works Great!"+"\\r\\n");
        out.print(
            "Thanks for the example - it works great!"+"\\r\\n");
        out.close();
        System.out.println("Message Sent");
    } catch (IOException e) {
        System.out.println("Send Failed");
        e.printStackTrace();
    }
}
```

20) What is the difference between a URI and a URL?

URLs are a subset of all URIs.

The term "Uniform Resource Locator" (URL) refers to the subset of URI that identify resources via a representation of their primary access mechanism (e.g., their network "location"), rather than identifying the resource by name or by some other attribute(s) of that resource.

## 21) How do I create a simple Datagram client?

1. First allocate space to hold the data we are sending and create an instance of DatagramPacket to hold the data.
2.     `byte[] buffer = new byte[1024];`
3.     `int port = 1234;`
4.     `InetAddress host = InetAddress.getByName("jguru.com");`
5.     `DatagramPacket packet = new DatagramPacket(buffer, buffer.length,`
6.                                     `host, port);`
7. Create a DatagramSocket and send the packet using this socket.
8.     `DatagramSocket socket = new DatagramSocket();`
9.     `socket.send(packet);`

The DatagramSocket constructor that takes no arguments will allocate a free local port to use. You can find out what local port number has been allocated for your socket, along with other information about your socket if needed.

```
// Find out where we are sending from
InetAddress localHostname = socket.getLocalAddress();
int localPort = socket.getLocalPort();
```

The client then waits for a reply from the server. Many protocols require the server to reply to the host and port number that the client used, so the client can now invoke socket.receive() to wait for information from the server.

## 22) How do I create a simple Datagram server?

To create a server with UDP, do the following:

1. Create a DatagramSocket attached to a port.
2.     int port = 1234;
3.     DatagramSocket socket = new DatagramSocket(port);
4. Allocate space to hold the incoming packet, and create an instance of DatagramPacket to hold the incoming data.
5.     byte[] buffer = new byte[1024];
6.     DatagramPacket packet =
7.         new DatagramPacket(buffer, buffer.length);
8. Block until a packet is received, then extract the information you need from the packet.
9.     // Block on receive()
10.    socket.receive(packet);
- 11.
12.    // Find out where packet came from
13.    // so we can reply to the same host/port
14.    InetAddress remoteHost = packet.getAddress();
15.    int remotePort = packet.getPort();
- 16.
17.    // Extract the packet data
18.    byte[] data = packet.getData();

The server can now process the data it has received from the client, and issue an appropriate reply in response to the client's request.

## 23) What is a port?

Port numbers are the means by which an operating system routes incoming packets to the appropriate waiting process. Only one process at a time can listen for incoming packets on a given port. The combination of destination IP address and port uniquely identifies the destination process for a packet. Likewise, the combination of source IP address and port uniquely identifies the source process.

For both TCP and UDP, the port number field of a packet is specified as a 16-bit unsigned integer - this means that valid port numbers range from 1 through 65535. (Port number 0 is reserved and can't be used).

Java does not have any unsigned data types; Java's short data type is 16 bits, but its range is -32768 to 32767 because it is a signed type. Thus, short is not large enough to hold a port number, so all classes which use or return a port number must represent the port number as an int. In the JDK 1.1+, using an int with a value greater than 65535 will generate an `IllegalArgumentException`. In the JDK 1.0.2 and earlier, values greater than 65535 are truncated and only the low-order 16 bits are used.

Port numbers 1 through 255 are reserved for well-known services. A well-known service is a service that is widely implemented which resides at a published, "well-known", port. If you connect to port 80 of a host, for instance, you may expect to find an HTTP server. On UNIX machines, ports less than 1024 are privileged and can only be bound by the root user. This is so an arbitrary user on a multi-user system can't impersonate well-known services like TELNET (port 23), creating a security problem. Windows has no such restrictions, but you should program as if it did so that your applications will work cross-platform.

24) How do I control whether `URLConnection` uses the POST or GET method when sending data to an HTTP server?

Since these methods are part of the HTTP protocol, they can only be used when the URL is an "http://" URL. In that case, you can cast the connection returned from `URLConnection.openConnection()` to be a `HttpURLConnection`, which has a `setRequestMethod()` method which lets you select whether GET or POST is used. For example:

```
if (connection instanceof HttpURLConnection)
    ((HttpURLConnection)connection).setRequestMethod("POST");
```

GET is used by default.

25) What is a proxy server and how does it work?

A proxy is something or somebody representing some other party. In the case of an HTTP proxy server, it is a server that relays browser requests to the appropriate web site, receives the web server response, and relays the response back to the browser. An HTTP proxy server is typically used through a firewall to avoid clients connecting directly to the Internet. The HTTP protocol explains how a client should get pages through a proxy.

26) What is a URI?

A URI (Uniform Resource Identifier) is a means of unambiguously locating a resource on the Internet.

There are two types of URIs:

- URL (Uniform Resource Locator)
- URN (Uniform Resource Name)

A URL is a pointer to a particular resource on the Internet at a particular location, while a URN is a pointer to a particular resource but without reference to a particular location.

27) Is there a way to programmatically find out if a port number is in use?

You can do the following:

```
try {  
    ServerSocket s = new ServerSocket(portno);  
}  
catch(IOException e) {  
    System.out.println("port " + portno + " already in use");  
}
```

28) How can I determine when multiple instances of my program are running on the same machine?

The most portable, general way is for your program to attempt to create a server socket on a particular port. The first instance will be able to create the port and so it will know that it's the first. Subsequent instances won't be able to bind to that server socket and so they'll know that they aren't the first.

If you need to track or otherwise manage all of the instances then you can change things such that if they can't bind to the server socket that they then try to connect to the first instance of the program using a client connect through the socket and you can do whatever you want from there.

Of course, if you're worried about controlling the number of copies of your program for "copy protection" purposes or you're wanting to control multiple instances of unsigned applets then you need to mediate the connections through a trusted server that you control.

29) How would you send a file over a socket connection?

If both ends of the connection knew exactly what was happening or expected (e.g. a single file with a constant filename, in the simplest case), you could do something as simple as have one end read from a `FileInputStream` and write to the socket's `OutputStream`, while the other end reads from the socket's `InputStream` and writes to a `FileOutputStream`.

If you need to transfer multiple files, or need to transfer some details about the file other than the contents (for example, the file name and path), you could

create a class which represents the file and serialize it out to the socket. At the simplest that might look like this:

```
import java.io.*;

public class TransferableFile implements Serializable
{
    private byte[] contents;
    private String fileName;

    public TransferableFile(String _fileName) {
        fileName = _fileName;
        // Open a FileInputStream and populate
        // the contents array
    }

    public void writeFile() {
        // Open a FileOutputStream to write
        // the contents array
    }
}
```

.. at one end you could create the TransferableFile and serialize it out to the socket. At the other end you deserialize and invoke writeFile().

An improvement on that would be to do the file I/O in the readObject() and writeObject() methods. That is, read the file from the disk in writeObject() and stream the data straight out to the socket, then write the data directly to the disk in readObject(). This would allow you to handle arbitrarily large files without changing the memory requirement.

If you need to do anything more complex than that, your best bet is to get an FTP client/server. A quick search in this Networking FAQ or in Google should list a few options.

### 30) What is "local loop back" used in networking

127.0.0.1 is a special IP address reserved for the local host, or "loopback address". It refers to the host on which your program is running. Operating systems typically implement "network" communications with the local host by a sort of software short-circuiting the ethernet hardware, effectively looping back the connection.

### 31) My computer is mapped to more than one IP address. From my program, how can I retrieve all the IP addresses the machine is mapped to?

You can use something like:



```
InetAddress ia=InetAddress.getLocalHost();  
InetAddress[] a=ia.getAllByName(ia.getHostName());
```

32) What is a Keep-Alive? How is it implemented differently in HTTP 1.0 and HTTP 1.1?

Http operates on what is called the request/response paradigm. The client application generates a request for information which is passed to the server, which then replies to it. In the earliest implementation of HTTP, each request created a new socket connection to the server, sent the command, then read the response from the same connection. Although this was simple to specify and implement, it was also slow, especially in a high volume situation.

Keep-alives were added to HTTP to basically reduce the significant overhead of rapidly creating and closing socket connections for each new request. The following is a summary of how it works within HTTP 1.0 and 1.1:

#### HTTP 1.0

The HTTP 1.0 specification does not really delve into how Keep-Alive should work. Basically, browsers that support Keep-Alive appends an additional header to the request as:

Connection: Keep-Alive

When the server processes the request and generates a response, it also adds a header to the response:

Connection: Keep-Alive

When this is done, the socket connection is not closed as before, but kept open after sending the response. When the client sends another request, it reuses the same connection. The connection will continue to be reused until either the client or the server decides that the conversation is over, and one of them drops the connection.

#### HTTP 1.1

Under HTTP 1.1, Keep-Alive is implemented differently. All connections, by default, are kept open, unless stated otherwise with the following header:

Connection: close

33) what is URLConnection?

URLConnection is a general purpose class for accessing the attributes of a remote resource. Once you make a connection to a remote server, you can use URLConnection to

inspect the properties of the remote object before actually transporting it locally. These attributes are exposed by the http protocol specification and as such only make sense for URL objects that are using the http protocol.

## Tomcat

### 1) How can I debug my servlet?

Hoo boy, that's a tough one.

First off, you should always do your own exception handling. An uncaught exception can silently kill your servlet, and if you don't know where to look in the log files, or if your server has a bug in it whereby it silently swallows certain exceptions, you'll have no idea where the trouble is.

The following code sets up a catch block that will trap any exception, and print its value to standard error output and to the ServletOutputStream so that the exception shows up on the browser (rather than being swallowed by the log file). Chances are that any error is in your code; the exception shows you what line the problem happened at. (If you see "Compiled Code" instead of line numbers in the exception stack trace, then turn off the JIT in your server.)

```
res.setContentType("text/html");
ServletOutputStream out = res.getOutputStream();
try {
    // do your thing here
    ...
}
catch (Exception e) {
    StringWriter sw = new StringWriter();
    PrintWriter pw = new PrintWriter(sw);
    e.printStackTrace(pw);
    out.println("
");
    out.print(sw.toString());
    out.println("
");
}
```

Lately, I've started catching all Throwables, just in case. I know this is bad form but what are you gonna do?

Next, you should make liberal use of the log() method, and you should keep your own log files. Again, don't trust the server to do the right thing. Also, printing the log after every request can help debugging because you can immediately see the output of your servlet without going into the server-side log files. (Another problem this avoids is that some servlet engines forget to flush their logs after

each request, so even if you go to the server, you won't see the most recent log messages.)

Here's some source code you can add to any HttpServlet that keeps an in-memory log:

```
StringBuffer logBuffer = new StringBuffer();

public void log(String s) {
    s = new Date().toString() + ": " + s;
    System.err.println(s);
    logBuffer.append(s);
    logBuffer.append("\n");
    super.log(s);
}
```

And here's some code that you can add to the end of your doGet or doPost method that prints out the entire log after each request:

```
out.println("<HR>\n<H3>Error log this session:</H3>\n<PRE>");
out.println(logBuffer.toString());
out.println("</PRE><P>");
```

Both of these should be disabled once you actually ship your servlet, but they can be very useful during development and debugging.

You should remember to use `servletrunner` (renamed "JSDK WebServer" with the JSDK version 2 -- run with the script `startserver`) to debug your servlet. It's a tool that ships with the JSDK that basically starts up a miniature web server that runs your servlet inside itself. It means you don't have to stop and restart your real web server every time you recompile your servlet. It also affords a more pure environment, so you can make sure your servlet truly conforms to the spec before you try to run it inside a (possibly buggy or nonstandard) servlet engine. (Note that Tomcat does not have a replacement for `servletrunner` :-(.)

A few IDEs support servlet debugging. Symantec Cafe claims to have a fairly robust system for doing visual source-level debugging of servlets (as well as RMI, CORBA, and EJB objects).

- 2) Can I use Tomcat or JavaWebServer as a service on Windows NT or Windows 2000?

For Tomcat:

Download `jk_nt_service.exe` and follow the instructions in the document "NT-Service-howto.html" (on the Jakarta Web site and included in the Tomcat distros). Also see the minimalistic users guide, included with the Tomcat distribution.

For JWS:

Near the end of the installation program you will be asked if you want to have the Java Web Server start automatically on system reboot. (That is, whether you want to install the Java Web Server as an NT Service).

If you click Yes: An entry will be added to the Control Panels -> Services and the JavaWebServer NT Service will be started up automatically every time you restart your system.

If you click No: No entry will be made in the Control Panel's Services panel.

If you change your mind later, the product documentation provides instructions for how to setup the web server to start automatically. For instructions, see the file:

`[server_root]\doc\en\administration\server_start_Win.html`

- 3) While I am still making changes to a servlet code, how can I make a servlet reload every time I test it?

It depends on the web server you are using to test your servlet. For instance, with Tomcat, you would replace the WAR file and the server notices the change and loads the new servlet on the next request, without having to restart the server.

[Short answer: for Tomcat, add `reloadable="true"` to the `<Context>` tag for that web application.

Note that there is a bug in Tomcat where replacing the WAR file doesn't work; you also have to delete the unpacked directory under `TOMCAT/webapps/foo` (for `TOMCAT/webapps/foo.war`).

If you are not using WAR file deployment, simply replacing the servlet classfile should work, as long as the class file is stored in `webapp/WEB-INF/classes`, or replacing the JAR file in `webapp/WEB-INF/lib/foo.jar`.

- 4) Is there any method to unload a servlet from Web Server memory without restarting the server?

There is no standard method/mechanism to unload a servlet from memory. Some servers, like JWS, provide the means to load and unload servlets from their

administration module. Others, like Tomcat, require you to just replace the WAR file.

- 5) I've just set up Tomcat 3.1, created a context, and when I try and run a sample JSP page, I get: Error: 500, Internal Servlet Error: java.lang.NullPointerException

You probably forgot to create a classes folder located at:

<context>/WEB-INF/classes

Create the folder, restart tomcat, all should be well.

- 6) How do I make Tomcat 3.1 invoke my servlets via the url <host>/servlets/<class> instead of <host>/servlet/<class> that it's doing now?

Change the url-pattern of the default servlet invoker located in the "deployment descriptor" for your web application. This file is usually located in <context>/WEB-INF/web.xml file. Replace:

[/servlet/\\*](#)

with:

[/servlets/\\*](#)

Also note that in Tomcat 3.1, web applications begin with the default "deployment descriptor" located in <tomcat home dir>/conf/web.xml. You can change the "default" for all web applications there.

- 7) Why doesn't Tomcat shut down when I run shutdown.bat (or shutdown.sh)?

One possibility is that if you disabled the Ajp12 connector in your server.xml you have also disabled the mechanism by which Tomcat shutdown is executed -- try putting that back in if you've commented it out

- 8) Where do I find web.xml and server.xml configuration files?

Note that Tomcat uses the default web.xml file, located in TOMCAT\_HOME/conf/web.xml, as the base for all contexts; individual webapps can override specific settings in a custom WEB-INF/web.xml file.

For server.xml, look in the [Tomcat distribution](#), in the directory you unzipped, under /doc/uguide/tomcat\_ug.html -- scroll down to the "Tomcat's Configuration Files" section.

- 9) How do I set init parameters in the servlet engine?

It depends on the servlet engine. Each has its own type of config file in which you can set the names and values of init parameters; these get passed to the servlets (or JSPs) via the `getInitParameter()` call.

For Tomcat and JSWDK, edit the [conf/web.xml](#) file.

10) Is Tomcat robust enough for production use?

IMHO yes.

We are presently running Tomcat (servlet & jsp requests) with Apache (graphics, perl, html) on 9 Solaris/Sparc servers servicing over 10 million servlet/jsp transactions per day.

11) Is there a problem with calling `Response.sendRedirect()` after `Response.addCookie()` ?

Yes, there is a bug in Tomcat 3.1 (and possibly other servlet engines). To send a cookie through a redirect, you must set the headers manually.

```
Cookie long_term = new Cookie(LONG_TERM_COOKIE_NAME,
    user_name);
    long_term.setMaxAge(60*60*24*4);
    long_term.setPath("/Blah");
    response.addCookie(long_term);
```

```
response.setStatus(HttpServletResponse.SC_MOVED_TEMPORARILY);
    response.setHeader("Location", REDIRECT_PAGE);
```

12) What version of the Servlets or JSP specification is supported by my favorite servlet product? (JSDK, JSWDK, Tomcat, J2EE, etc.)

Product	Product Version	Servlet spec	JSP spec
<a href="#">Tomcat</a>	3.1	2.2	1.1
<a href="#">Tomcat</a>	3.0	2.2	1.1
<a href="#">Sun J2EE Reference Implementation</a>	beta	2.2	?
<a href="#">Sun JSWDK</a>	1.0.1	2.1	1.0.1
<a href="#">Sun JSDK</a>	2.1	2.1	none
<a href="#">Sun JSDK</a>	2.0	2.0	none
<a href="#">Allaire JRun</a>	3.0	2.2	1.1
<a href="#">Sun Java Web Server</a>	2.0	2.1	-

<a href="#">Sun Java Web Server</a>	1.1	2.0	-
<a href="#">Sun/Netscape iPlanet Web Server</a>	4.1	2.2	?

13)Where should I place my beans and servlets under Tomcat?

Where you place the beans and servlets depends on which "application" it belongs to. Assuming that you are running your JSP pages under the default "examples" application, you should place your servlets under the path

/tomcat/examples/WEB-INF/classes

Your beans should be placed under

/tomcat/examples/WEB-INF/jsp/beans

Both these paths are placed within CLASSPATH by Tomcat's startup script file.

14)Can I configure Tomcat so that the name of the webapp is not used in the URL?

E.g. <http://localhost/servlet/HelloWorld>, instead of <http://localhost/examples/servlet/HelloWorld>.

You can ... but I'm not sure I'd do this on a live system!

The path you give a context must be unique within the server.xml document. Therefore, all you need to do is change the path of the ROOT webapp (to "/root" for example) and change the path of your webapp to an empty string as below.

```
<Context path="" docBase="webapps/examples" debug="0" reloadable="true" >
```

15)Can I use Tomcat under Windows 98? Is there anything I have to watch out for during configuration?

Tomcat works very well with win95/win98. Generally there is just one problem, in that, calling startup.bat or tomcat.bat results in very strange outputs. This is related to a space characters problem within the .bat file due to difference in file formats under win95 and winNT.

For all these problems, my standard solution was to copy the file with an editor like MS Word and save to another bat file. If you starts changing bat file yourself, its very confusing.

16)Can I use Basic HTTP Authentication using Apache+JServ?

I recently did this with Apache and Tomcat. Using Basic HTTP Authentication is an Apache function, so this method should work with Apache and any servlet engine. Different web servers will, of course, work differently.

If you haven't already done so, read the Apache FAQ about authentication (section G) at [apache.org](http://apache.org), especially QA G.2. Also read the Apache Week article referenced there (<http://www.apacheweek.com/issues/96-10-18#userauth>). These resources will give you a good idea about how Apache can be configured to restrict access to URL's. Neither one explicitly tells you how to use authentication with servlets, so I'll spell it out here.

Use the <Location> directive to indicate to Apache that your specific servlet URL or servlet URL prefix (for multiple servlets) can be accessed only by authenticated users. The following template should be added to one of the Apache configuration files (such as http.conf) with appropriate substitutions for your system:

```
<Location /your/servlet/url >  
  AuthName "your realm"  
  AuthType Basic  
  AuthUserFile /your/user/file  
  require valid-user  
</Location>
```

This <Location> directive tells Apache to restrict access to the specified URL, so don't use the filesystem path to your servlet. Use the servlet's URL.

You will also need to create a user file with htpasswd.

- 17) With regard to version 3.1 of Tomcat, what specifically must be done to enable automatic reloading of servlets?

Automatic reloading of servlets by a servlet container comes in handy during the development phase when you are frequently modifying and recompiling the servlets. Having to constantly shutdown and restart the servlet container each time you modify a servlet class can become tiring very quickly.

Automatic reloading of servlets in Tomcat 3.1 is predicated upon two entries in two different files in the directory TOMCAT\_HOME/conf:

1. In the file TOMCAT\_HOME/conf/server.xml, you must set the reloadable attribute to true in the context element pertaining to your webapp. For example, I have a webapp named form-processing. So I added the following entry to the file TOMCAT\_HOME/conf/server.xml:
- 2.
3.     <Context path="/form-processing" docBase="webapps/form-processing"
4.         debug="0" reloadable="true" >
5.     </Context>
- 6.



7. The file `TOMCAT_HOME/conf/tomcat.properties` allows you to specify a classpath that is passed on to the JVM. The servlets that you want automatically reloaded must not be in this classpath. Note that this classpath is in addition to the classpath specified in the startup file `tomcat.bat` in the directory `TOMCAT_HOME/bin`. For the simple webapp form-processing that I mentioned above, I do not need to specify any additional classpaths in the `TOMCAT_HOME/conf/tomcat.properties` file.

For those just getting into Tomcat 3.1, I should add the caveat mentioned in the Tomcat readme file that the automatic reload feature in Tomcat 3.1 is not recommended for production applications. At this time, it is an experimental feature that when enabled creates extra overhead in processing servlet requests.

- 18) I had Tomcat installed on `c:\tc\jakarta-tomcat`, but need to run `.jsp` stored on `d:\abc\test`. How can I accomplish this?

Unless you are using a "J2EE web application" there is no simple way to do this. I would suggest creating a new web application and registering it with tomcat.

A web application is essentially a standardised directory structure containing all of the resources for your application. Using this standardised approach means that you can migrate your application to another JSP 1.1/Servlet 2.2 compatible server in the future very easily.

The examples directory under your tomcat webapps directory (`c:\tc\jakarta-tomcat\webapps`) is a good example of how to setup a webapp, and the important points to note are the `WEB-INF` directory and its `web.xml` file.

Once you've setup your web application, you can register it with tomcat by opening up the `server.xml` file from the "conf" directory under your tomcat installation. Again, take a look at how the example application is registered and use the same mechanism for registering your new webapp. All you need to do is specify the "docBase" attribute as a fully qualified path.

- 19) If I enable automatic servlet reloading for a webapp, will the `init()` method of the servlet get executed for each automatic reload?

Yes.

This question is important because it is possible to misconstrue the community literature dealing with servlets and believe that the `init()` method of a servlet will get executed only once -- the first time a servlet gets loaded into the container.

The following simple servlet, called `TestServlet`, can be used to verify the answer to the question. In my Tomcat 3.1 container, this servlet is inside a webapp called `test-suite` and resides in the directory

`TOMCAT_HOME/webapps/test-suite/WEB-INF/classes`

In accordance with the jGuru FAQ [posting](#), I have enabled automatic reloading for this webapp by including the following element in the server.xml file in the TOMCAT\_HOME/conf directory:

```
<Context path="/test-suite" docBase="webapps/test-suite" debug="0"
reloadable="true" > </Context>
```

As the following code shows, the servlet contains a line "Revision number: xx" where xx can be edited to create a change in the servlet to trigger automatic reload. Upon each reload, the print statement in the init() method will be printed out in the Tomcat console window. Here is the code for this servlet:

```
//filename: TestServlet.java
```

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
```

```
public class TestServlet extends HttpServlet {

    public void init() throws ServletException
    {
        System.out.println("TestServlet being loaded into the container");
    }

    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException
    {
        response.setContentType( "text/html" );
        PrintWriter out = response.getWriter();

        out.println( "<html>" +
            "<head><title> TestServlet </title></head>" +
            "Hello there from TestServlet ----- " +
            "Revision number: 18" +
            "</body></html>" );
    }
}
```

## 20)What is Tomcat?

Tomcat is the code name for the flagship product of the [Jakarta Project](#) . It is a commercial-quality, open-source implementation of the Java Servlet 2.2 and JavaServer Pages 1.1 Specifications that runs either standalone or integrated with a

web server. It is also the official Reference Implementation for those specifications (sanctified by Sun).

21) With regard to automatic servlet reloading in Tomcat 3.1, what happens if one servlet makes references to other servlets in the same webapp? Will automatic reloading get enabled for all such servlets?

The short answer to the question is: No.

Let's say that a webapp for which automatic reload has been enabled consists of a group of mutually referencing servlets. Let's also say that you have modified and recompiled all the servlets. When you hit the reload button on the browser, only one servlet -- the servlet that the browser is pointing to -- will be automatically reloaded by the container. Hitting the reload button on the browser for the other servlets will not cause their reload into the container and you will only see their older versions. If you want the newer versions of the other servlets to be reloaded into the container, in general you'd need to shutdown and restart the servlet container.

This behavior of Tomcat 3.1 is in keeping with the following statement that appears in Tomcat 3.1 [release notes](#):

*"..... changes to classes other than the servlet you are requesting do not trigger class reloads -- you will need to restart Tomcat to reflect changes in those classes."*

In the rest of this posting, I will explain this property of Tomcat 3.1 with the help an example. The reader is also referred to a [posting](#) by Costin Manolache on this aspect of automatic class reloading in Tomcat 3.1.

To demonstrate this behavior of Tomcat 3.1, I have constructed the following example that consists of two simple mutually referencing servlets called TestServlet\_1 and TestServlet\_2. In my Tomcat 3.1 container, both these servlets are in a webapp called test-suite and reside in the directory TOMCAT\_HOME/webapps/test-suite/WEB-INF/classes. In accordance with the jGuru FAQ [posting](#), I have enabled automatic reloading for this webapp by including the following element in the server.xml file in the TOMCAT\_HOME/conf directory:

```
<Context path="/test-suite" docBase="webapps/test-suite" debug="0"
        reloadable="true" > </Context>
```

Let's say that after starting the servlet container, you point your browser to the following URL:

[http://localhost:8080/test-suite/servlet/TestServlet\\_1](http://localhost:8080/test-suite/servlet/TestServlet_1)

the servlet container will load TestServlet\_1 into the container, which will cause the browser to display a "Hello..." message. If you click inside this message where indicated, the browser will make a request for TestServlet\_2 and it will be loaded into the container.

Now suppose you make changes to both the servlets by altering, say, the value of "Revision number: ". If your browser is pointing to TestServlet\_1 and you hit reload button, the servlet container will automatically reload TestServlet\_1. But the container will NOT reload TestServlet\_2 even if you hit the reload button on the browser when the browser is pointing to TestServlet\_2.

To see the newer version of TestServlet\_2, you have two choices: 1) Recompile the classes and this time start out by pointing your browser to TestServlet\_2. Or, 2) Shutdown and restart the servlet container.

Here is the code for the two servlets:

```
////////// file: TestServlet_1.java //////////
```

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
```

```
public class TestServlet_1 extends HttpServlet {

    public void init() throws ServletException
    {
        System.out.println("TestServlet_1 being loaded into the container");
    }

    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException
    {
        response.setContentType( "text/html" );
        PrintWriter out = response.getWriter();

        out.println( "<html>" +
            "<head><title> TestServlet_1 </title></head>" +
            "Hello there from TestServlet_1 ----- " +
            "To see the hello message from TestServlet_2," +
            " click <a href=\"/test-suite/servlet/TestServlet_2\">here</a> ----- " +
            "Revision number: 17" +
            "</body></html>" );
    }
}
```

```
////////// file: TestServlet_2.java //////////
```

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
```

```
public class TestServlet_2 extends HttpServlet {

    public void init() throws ServletException
    {
        System.out.println("TestServlet_2 being loaded into the container");
    }

    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException
    {
        response.setContentType( "text/html" );
        PrintWriter out = response.getWriter();

        out.println( "<html>" +
            "<head><title> TestServlet_2 </title></head>" +
            "TestServlet_2 saying hello ----- " +
            "To see the hello message from TestServlet_1," +
            " click <a href=\"/test-suite/servlet/TestServlet_1\">here</a> ----- " +
            "Revision number: 17" +
            "</body></html>" );
    }
}
```

22) Is there any way to tell Tomcat (or any servlet engine) never to use cookies for session management?

I'm not sure about Tomcat but in WebLogic (from 4.0.2 to 5.1 versions) there is parameter in the weblogic.properties file which can enable (or disable) cookies.

The parameter line is:

weblogic.httpd.session.cookies.enable=true

or

weblogic.httpd.session.cookies.enable=false

23) Is it possible to use a TagLib without defining a web app using Tomcat? [I'm using .jsp without webapps and everything's ok, but I can't make it work with taglibs]

When using taglibs with an application you specify the location of a tag library in the web.xml file, web.xml resides in TOMCAT\_ROOT/webapps/APPNAME/WEB-INF. The taglib entries in web.xml look like:

```
<taglib>
  <taglib-uri>
    blahblah
  </taglib-uri>
  <taglib-location>
    /WEB-INF/jsp/BlahBlah.tld
  </taglib-location>
</taglib>
```

There is another web.xml which seems to be a global one: TOMCAT\_ROOT/conf/web.xml. Simply place the above XML there, the last entry in my conf/web.xml was a "welcome-file-list" entry, the taglib xml can be placed after that. I don't know where the .tld file should be placed, maybe under TOMCAT\_ROOT/webapps/ROOT/WEB-INF/jsp ?

24) Does Tomcat support EJB?

No, Tomcat does not support EJB. It is a Servlet/JSP container, not an EJB container.

The [Jakarta Project](#) provides an implementation of the Servlets API called Tomcat, and an implementation of JavaServer Pages called Jasper. These are bundled together in the Tomcat release.

However, the [J2EE Reference Implementation](#), which supports EJB, is bundled with an old version of Tomcat. This adds Tomcat's support for Servlets and JSP to the J2EE product.

25) How do I deploy a WAR file in Tomcat?

Currently (as of version 3.2) the procedure for deploying a new WAR file is:

1. Stop Tomcat.
2. Delete existing deployment. If you have previously deployed "foo.war" in TOMCAT\_HOME/webapps, then it has been unpacked into

webapps/foo/... You must delete this directory and all its contents. On Unix, this can be done with

```
rm -r $TOMCAT_HOME/webapps/foo
```

3. Copy WAR file to TOMCAT\_HOME/webapps/.
4. Start Tomcat.

This process may become easier in the future. A "deploy tool" is on the Tomcat "to-do" list.

Note that if you deploy a WAR file in this manner, you do not need to make any changes to server.xml -- it will automatically be recognized and activated. However, if you wish to specify non-default options for this webapp, you may do so by adding an element like

```
<Context docBase="webapps/foo" ...
```

to server.xml.

26) What is the difference between JServ and Tomcat? And what's up with mod\_jserv and mod\_jk?

[JServ](#) was one of the very first servlet engines. It is now an open source project, consisting of two parts: a pure-java servlet engine, compliant with the Servlet spec version 2.0, and an Apache module written in C called mod\_jserv that allows Apache to send servlet requests to the JServ JVM.

[Tomcat](#) is a completely separate product, originally written by Sun, that is now also open-source and administered by Apache. It supports the Servlet spec version 2.2, and the JSP spec version 1.1. It also contains a pure Java servlet engine, and a number of connectors for several commercial web servers.

Here's the confusing part: originally, when it was time to write a connector from Apache to Tomcat, they started with the code base of mod\_jserv. Although it has been rewritten and revised to the point where it is no longer compatible with the mod\_jserv that comes with JServ, unfortunately, it is still called mod\_jserv. This means that if you have the JServ mod\_jserv installed, it will not work with Tomcat, and vice versa.

Fortunately, the latest version of Tomcat comes with a new connector for Apache, called mod\_jk. This connector supports Tomcat but not JServ. It is also more advanced, in that it supports Tomcat-style load balancing, and also supports SSL authentication (though this latter functionality is still a bit rough and may have bugs in actually \*reporting\* that the connection is secure).

So if you just use mod\_jk and Tomcat, you should have no problems (at least as far as this connector issue is concerned).

27) I am running Tomcat on Apache for NT. When I put a \ after the jsp page, e.g. "http://www.myserver.com/includes/top\_nav.jsp\", it shows the code, i.e. the java embedded in the HTML. How do I resolve this issue?

This is a bug. AFAIK it has been fixed in the latest version (3.2) of Tomcat.

28) Why do I get a "java.net.BindException:Address in use" error?

Probably because the address is in use :-)

You're already running an instance of your servlet engine, or another service is running on its port. Make sure the other service is fully stopped before launching another one.

29) What are the contents of WEB-INF?

The contents of the WEB-INF directory are:

- /WEB-INF/web.xml deployment descriptor
- /WEB-INF/classes/\* directory for servlet and utility classes. The classes in this directory are used by the application class loader to load classes from.
- /WEB-INF/lib/\*.jar area for Java ARchive files which contain servlets, beans, and other utility classes useful to the web application. All such archive files are used by the web application class loader to load classes from.

<h3>Sample Web Application Directory Structure</h3>

Illustrated here is a listing of all the files in a sample web application:

```
/index.html
/htdocs.jsp
/feedback.jsp
/images/banner.gif
/images/jumping.gif
/WEB-INF/web.xml
/WEB-INF/lib/jspbean.jar
/WEB-INF/classes/com/mycorp/servlets/MyServlet.class
/WEB-INF/classes/com/mycorp/util/MyUtils.class
```

30) What is a WAR file and how do I create one?

A WAR (or "web archive") file is simply a packaged webapp directory. It is created using the standard Java jar tool. For example:

```
cd /home/alex/webapps/mywebapp
jar cf ../mywebapp.war *
```



### 31) How are initialization parameters passed to a servlet under Servlet API version 2.2?

Through the webapp deployment descriptor, which is the web.xml file in the directory WEB-INF.

Note that there are two different types of web.xml files in Tomcat 3.x: There is a web.xml that sits in the top-level configuration directory TOMCAT\_HOME/conf and then there can be a separate web.xml for each webapp in the directory WEB-INF for that webapp. You would want to keep webapp specific deployment information, such as initialization parameters and their values, in the web.xml specific to that webapp.

The best on-line example that illustrates servlet initialization through a webapp's deployment descriptor is, I believe, the ServletParam servlet in the test webapp in your Tomcat software package. This servlet can be called with two different canonical names, as for example in

`http://localhost:8080/test/servlet/servletParam1`

and

`http://localhost:8080/test/servlet/servletParam2`

The deployment descriptor maps the canonical names servletParam1 and servletParam2 to the actual servlet ServletParam but with different values of the initialization parameters. When the servlet ServletParam is called with the name servletParam1, the initialization parameter-value pairs are

param1 value1  
param2 value2

and when the same servlet is called with the name servletParam2, the parameter-value pairs are

param3 value3  
param4 value4

The following constitutes a sufficient deployment descriptor for this example:

```
<!-- filename: web.xml in the directory TOMCAT_HOME/webapps/<your-  
webapp>/WEB-INF/ -->
```

```
<?xml version="1.0"?>
```

```
<!DOCTYPE web-app SYSTEM "http://java.sun.com/j2ee/dtds/web-app_2.2.dtd">
```

```
<web-app>
  <servlet>
    <servlet-name>
      servletParam1
    </servlet-name>
    <servlet-class>
      ServletParam
    </servlet-class>
    <init-param>
      <param-name>param1</param-name>
      <param-value>value1</param-value>
    </init-param>
    <init-param>
      <param-name>param2</param-name>
      <pram-value>value2</pram-value>
    </init-pram>
  </servlet>

  <servlet>
    <servlet-name>
      servletParam2
    </servlet-name>
    <servlet-class>
      ServletParam
    </servlet-class>
    <init-pram>
      <pram-name>param3</pram-name>
      <pram-value>value3</pram-value>
    </init-pram>
    <init-pram>
      <pram-name>param4</pram-name>
      <pram-value>value5000</pram-value>
    </init-pram>
  </servlet>
</web-app>
```

The first half of the web.xml file associates the servlet ServletParam with the name servletParam1 and then goes on to declare its parameter-value pairs. The second half does the same for the name servletParam2.

32) Is it possible to bind Tomcat to one specific IP address on a multi-homed host (for virtual hosting)?

Yes, from version 3.1 you can a virtual host bind to one of the ip addresses owned by the machine, if you add something like the following to server.xml:

```
<Host name="127.0.0.1" >
  <Context path=""
    docBase="webapps/examples" />
  <Context path="/examples"
    docBase="webapps/ROOT" />
</Host>
```

as you can see you can put a IP adress for the Virtual Host name, ergo you can have a Virtual Host pointed to any particular IP configured in your machine.

33) Can I use the latest version of Tomcat inside Sun's J2EE implementation?

The J2EE reference implementation ships with Tomcat. The classes for Tomcat are located inside the \lib\j2ee.jar. If you wanted to use the latest version of Tomcat you would have to replace the classes in this archive.

This would of course not be supported.

34) What are the advantages/disadvantages of using Tomcat's built-in HTTP server instead of, say, Apache?

Regarding Tomcat's disadvantages when used as a stand-alone server, here is a list reproduced from the [Tomcat-Apache HOWTO](#) page:

1. Tomcat is not as fast as Apache when it comes to static pages.
2. Tomcat is not as configurable as Apache.
3. Tomcat is not as robust as Apache.
4. There are many sites with long time investment in certain web servers, for example, sites that are using CGI scripts/Server API modules/perl/php. We cannot assume that all of them will want to ditch this legacy.

The page further says that " For all these reasons it is recommended that real world sites use an industrial strength web server, such as Apache, for serving the static content of the site, and use Tomcat as a Servlet/JSP add-on."

The main thing that Tomcat has going for it is that it the official reference implementation of the Java Servlet 2.2 and JavaServer Pages 1.1 technologies developed under the Apache license by some of the best developers around the world.

35) What is an invoker servlet in a Tomcat servlet container?

The "invoker servlet" is a system-supplied servlet that extracts the name of the requested servlet class from the portion of the URL that comes after "/servlet".

The invoker servlet then loads the requested servlet into the container and executes it.

If you look at the contents of the webapp deployment descriptor file, web.xml, in the TOMCAT\_HOME/conf directory, you will find the following servlet element at the beginning:

```
<servlet>
  <servlet-name>
    invoker
  </servlet-name>
  <servlet-class>
    org.apache.tomcat.servlets.InvokerServlet
  </servlet-class>
</servlet>
```

which declares the name and the place of the system-supplied invoker servlet. Shortly thereafter appears the following servlet-mapping element:

```
<servlet-mapping>
  <servlet-name>
    invoker
  </servlet-name>
  <url-pattern>
    /servlet/*
  </url-pattern>
</servlet-mapping>
```

which tells the container that all requests that begin with "/servlet" (relative to root of the webapp specified by the name that comes before "/servlet") are to be sent to the invoker servlet.

37) When I recompile a servlet which is already loaded by the container, it is automatically reloaded. However the destroy() method of the servlet does not get called. Does this mean that the servlet is not unloaded before reloading?

The observation you make does not apply to the version of Tomcat 3.1 that is currently available for download from the Jakarta-Tomcat site. However, it does apply to Tomcat 3.2 beta 4. More on 3.2 later in this posting. First, I'd like to say a few words about what I am basing my 3.1 answer on.

The test servlet for which I show code below has both an `init()` method and a `destroy()` method. When the `init()` method is invoked, it prints out the following message in the Tomcat window:

TestServlet being loaded into the container

and when the `destroy()` method is invoked, it prints out the following message in the Tomcat window:

TestServlet about to be unloaded

Note that the `init()` method will only be invoked when the servlet class is loaded into the container and its instance created. The `destroy()` method will only be invoked just prior to the container unloading the servlet.

If I carry out the following steps in the sequence indicated:

1. request the servlet through a client browser
2. recompile the servlet
3. hit the reload button on the browser

I get the following sequence of messages in the Tomcat window:

TestServlet being loaded into the container

TestServlet about to be unloaded

RELOAD!!!

TestServlet being loaded into the container

This shows the servlet is actually unloaded and its `destroy()` method invoked before the new version of the servlet is automatically loaded in.

Here is the code for the servlet:

```
import java.io.*;
```

```

import javax.servlet.*;
import javax.servlet.http.*;

public class TestServlet extends HttpServlet {

    public void init() throws ServletException
    {
        System.out.println("TestServlet being loaded into the container");
    }

    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException
    {
        response.setContentType( "text/html" );
        PrintWriter out = response.getWriter();

        out.println( "<html>" +
            "<head><title> TestServlet </title></head>" +
            "Hello there from TestServlet" +
            "</body></html>" );
        out.close();
    }

    public void destroy()
    {
        System.out.println( "TestServlet about to be unloaded" );
    }
}

```

Therefore, with regard to automatic loading and unloading, Tomcat 3.1 works as specified in the Servlet 2.1 API. However, the beta 4 version of Tomcat 3.2 seems to have a bug in it. When you do exactly the same thing as described above in Tomcat 3.2, the `destroy()` method is not invoked, meaning that the old copy of the servlet does not get unloaded.

38) I've setup Tomcat 3.1 with Apache on my web server to serve JSP pages. It works well for Java Servlets and JSP pages. However, this is one issue: if the target JSP page does not exist, the server will send out a File Not Found Error

404 along with the DETAIL PATH, such as "File /usr/local/jakarta-tomcat/webapps/... Not found". How can I stop tomcat from spilling off its own detail file path on the server.

If you look the code of the Tomcat's JSPServlet.java, you will find the following code in the service method:

```
try{
//look for the jsp file
} catch (FileNotFoundException ex) {
response.sendError(HttpServletResponse.SC_NOT_FOUND,
Constants.getString("jsp.error.file.not.found", new Object[] { ex.getMessage() }));
```

As you can see that the FileNotFoundException directly works with the response object and the message is taken from the messages.properties files to send it directly to the caller (browser). Also I did not find any other mechanism in the Tomcat's configuration to suppress the error messages. So it will not be possible to suppress this message.

39) How is SingleThreadModel implemented in Tomcat? In other containers? [I would assume that Tomcat uses its connection thread pool, and creates a new instance of the servlet for each connection thread, instead of sharing one instance among all threads. Is that true?]

The question mixes together two rather independent aspects of a servlet container: "concurrency control" and "thread pooling".

Concurrency control, such as achieved by having a servlet implement the SingleThreadModel interface, addresses the issue of thread safety. A servlet will be thread-safe or thread-unsafe regardless of whether the servlet container used a thread pool. Thread pooling merely eliminates the overhead associated with the creation and destruction of threads as a servlet container tries to respond to multiple requests received simultaneously. It is for this reason that the specification document for Servlet 2.2 API is silent on the subject of thread pooling -- as it is merely an implementation detail. However, the document does indeed address the issue of thread safety and how and when to use SingleThreadModel servlets.

Section 3.3.3.1 of the Servlet 2.2 API Specification document says that if a servlet implements the SingleThreadModel it is guaranteed "that only one request thread at time will be allowed in the service method." It says further that "a servlet container may satisfy this guarantee by serializing requests on a servlet or by maintaining a pool of servlet instances."

Obviously, for superior performance you'd want the servlet container to create multiple instances of a SingleThreadModel type servlet should there be many requests received in quick succession. Whether or not a servlet container does that depends completely on the implementation. My experiments show that

Tomcat 3.1 does indeed create multiple instances of a SingleThreadModel servlet, but only for the first batch of requests received concurrently. For subsequent batches of concurrent requests, it seems to use only one of those instances.

40) Which servlet containers have persistent session support? Specifically, does Tomcat 3.1?

All servlet containers that implement the Servlet 2.2 API must provide for session tracking through either the use of cookies or through URL rewriting. All Tomcat servlet containers support session tracking.

41) How do I write to a log file using JSP under Tomcat? Can I make use of the log() method for this?

Yes, you can use the Servlet API's log method in Tomcat from within JSPs or servlets. These messages are stored in the server's log directory in a file called servlet.log.

42) When Using Tomcat 3.1 with Apache, how do you set up your context and web server index so that a .jsp file is set as the index file?

In Apache, first map the desired jsp filename to the directory index. For example:

In srm.conf [or httpd.conf]:

DirectoryIndex index.jsp

Then in your webapp's web.xml or in your server's web.xml, look at the bottom for the <welcome-file-list> tag. You can add <welcome-file> tags to this tag, such as:

```
<welcome-file-list>
  <welcome-file>
    desired-index.jsp
  </welcome-file>
</welcome-file-list>
```

43) Can I include two jsp:plugin actions on one JSP page? It does not compile under Tomcat

Yes. Just include an extra set of {}'s around the plugin code:

```
<% { %>
<jsp:plugin
  type="applet"
  code="Clock2.class"
  codebase="/examples/jsp/plugin/applet"
  jreversion="1.2"
```



```

width="160"
height="150" >
  <jsp:fallback>
    Plugin tag not supported.
  </jsp:fallback>
</jsp:plugin>
<% } %>
<% { %>
<jsp:plugin
  type="applet"
  code="Clock2.class"
  codebase="/examples/jsp/plugin/applet"
  jreversion="1.2"
  width="160"
  height="150" >
    <jsp:fallback>
      Plugin tag not supported.
    </jsp:fallback>
  </jsp:plugin>
<% } %>

```

#### 44) How do you enable BASIC authentication using Tomcat?

[Question continues: I'm able to get the dialog box requesting the username and password to pop up, however, even though I insert one of the usernames and passwords from the tomcat-users.xml file, I always get an "Authorization failed. Retry?" dialog.]

Make sure that the user has been designated a role in the tomcat-users.xml that matches the role(s) defined in the auth-constraint section of the web.xml:

*web.xml:*

```

<auth-constraint>
<role-name>role1</role-name>
</auth-constraint>

```

*tomcat-users.xml:*

```

<tomcat-users>
<user name="test" password="tomcat" roles="role1" />
</tomcat-users>

```

#### 45) How do I set up virtual hosting on Tomcat in standalone mode?

In Tomcat 3.2b6, it is very easy to setup virtual hosts in standalone mode. First, be sure to set the correct port for the HTTP listener (the default is 8080, and you'll probably want to change it to 80). Second, wrap your servlet context definitions (in your server.xml) with a <host> tag, specifying the domain name to work on. Here is an excerpt from a server.xml that is configured for virtual hosting:

```

<Host name="server1.domain.com">
<Context path="" docBase="/var/www/server1" reloadable="true" debug="0" />
</Host>
<Host name="server2.domain.com">
<Context path="" docBase="/var/www/server1" reloadable="true" debug="0" />
</Host>

```

#### 46) How do I set up virtual hosting on Tomcat with Apache?

You can configure your Apache server's virtual hosts to map to virtual hosts in a single Tomcat instance. Each virtual host would exist as a separate servlet context.

[For mod\_jk instructions, see the user's guide. For mod\_jserv...]

First configure Apache with mod\_jserv, mapping the appropriate hostnames to Tomcat. Here is a sample from httpd.conf:

```

ApJServManual      on
ApJServSecretKey    DISABLED
ApJServMountCopy    on
ApJServLogLevel     notice

ApJServDefaultProtocol ajpv12
ApJServDefaultPort  8007
ApJServDefaultHost  localhost

AddType    text/jsp .jsp
AddHandler jserv-servlet .jsp

NameVirtualHost 192.168.0.2

<VirtualHost 192.168.0.2>
  DocumentRoot /var/www/server1
  ServerName server1.lokitech.com
  ApJServMount /servlet /ROOT
  <Directory /var/www/server1/WEB-INF>
    Order allow,deny
    Deny from all
  </Directory>
</VirtualHost>

<VirtualHost 192.168.0.2>
  DocumentRoot /var/www/server2
  ServerName server2.lokitech.com
  ApJServMount /servlet /ROOT
  <Directory /var/www/server2/WEB-INF>
    Order allow,deny

```

```
    Deny from all
  </Directory>
</VirtualHost>
```

Second, configure Tomcat for virtual hosts. In your server.xml, you will wrap each servlet context in a <host>. Here is a sample from server.xml:

```
<Host name="server1.domain.com">
  <Context path="" docBase="/var/www/server1" reloadable="true" debug="0" />
</Host>
<Host name="server2.domain.com">
  <Context path="" docBase="/var/www/server1" reloadable="true" debug="0" />
</Host>
```

47) How do you do servlet aliasing with Apache and Tomcat?

Servlet aliasing is a two part process with Apache and Tomcat. First, you must map the request in Apache to Tomcat with the ApJServMount directive, e.g.,

```
ApJServMount /myservlet /ROOT
```

Second, you must map that url pattern to a servlet name and then to a servlet class in your web.xml configuration file. Here is a sample excerpt:

```
<servlet>
  <servlet-name>myservlet</servlet-name>
  <servlet-class>com.mypackage.MyServlet</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>myservlet</servlet-name>
  <url-pattern>/myservlet</url-pattern>
</servlet-mapping>
```

48) Can Tomcat be configured to interpret all, or selected, .html files within a given context as JSP? Or, do JSP files have to end with a .jsp extension?

yes you can do that by modifying the web.xml file. You will have to invoke the org.apache.jasper.runtime.JspServlet for all the requests having extension .html. You can do that by changing the Servlet mapping code:

```
<servlet-mapping>
  <servlet-name>
jsp
  </servlet-name>
  <url>*.html</url>

</servlet-mapping>
```

And comment out the following block

```
<mime-mapping>
<extension>
html
</extension>
<mime-type>
text/html
</mime-type>
</mime-mapping>
```

49) Is there any way I can assign different values of CLASSPATH to different contexts?

One way to get the functional equivalent is to put the files in question in either "<app-path>/WEB-INF/classes" or "<app-path>/WEB-INF/lib", depending on whether you are talking about a .class or .jar. The CLASSPATH for a given webapp will include WEB-INF/classes and all the .jar files found in WEB-INF/lib. Say you had utils.jar and form-taglib.jar in WEB-INF/lib, the resulting classpath would look like:

```
$CLASSPATH:<app-path>/WEB-INF/classes:<app-path>/WEB-
INF/lib/utils.jar:<app-path>/WEB-INF/lib/form-taglib.jar
```

50) How do I change the port from 8080 to 80?

Open server.xml in %TOMCAT\_HOME%\conf directory and edit the <Connector> element.

For SimpleTcpConnector, change the sub-element "port" value to 80. Yes, You got it!

51) How to deploy a servlet that includes javax.mail.\* on Tomcat?

Just place the mail.jar and activation.jar files (and possibly the pop.jar file if you are using POP3) in the CLASSPATH of the web server. The simplest way to do this is to rely on the Java extensions framework. Add them to the JAVA\_HOME/jre/lib/ext directory. Just copy the files there. Otherwise, set the CLASSPATH environment variable before starting Tomcat. The startup scripts retain the existing CLASSPATH, adding the Tomcat-specific stuff to the end.

52) What is the difference between apache webserver, java webserver and tomcat server?

Apache is an HTTP server written in C that can be compiled and run on many platforms. See <http://www.apache.org/>

Java WebServer is an HTTP server from Sun written in Java that also supports Servlets and JSP.

Tomcat is an open-source HTTP server from the Apache Foundation, written in Java, that supports Servlets and JSP. It can also be used as a "plug-in" to native-

code HTTP servers, such as Apache Web Server and IIS, to provide support for Servlets (while still serving normal HTTP requests from the primary, native-code web server).

53) What is Catalina?

Catalina is the name given to the Tomcat 4.0 servlet engine component, in the same way that the JSP compiler/runtime component is called Jasper.

54) Where do I install the JDBC driver ZIP files, JAR files & DLLs? How do I ensure they're in Tomcat's CLASSPATH?

Since version 3.2, all the JARS present in the %TOMCAT\_HOME%/lib directory are automatically appended to Tomcat's classpath, but only the jars are appended. To have a ZIP automatically appended rename it to \*.jar ( as a jar is only a renamed zip ) and voila....

[There are two other options:

1. Put the jar file in your webapp's WEB-INF/lib/ directory. This set of JARs is added to the classpath for your webapp only. Be careful, however: if a different driver for the same JDBC URL is already on the system classpath, then it may (may!) be loaded instead of yours. (Whether it is or not depends on whether that driver has already been loaded yet.)
2. Change the Tomcat CLASSPATH environment variable, or "-classpath" command-line option, to contain your JAR or ZIP. This is set in the Tomcat startup script, tomcat.bat/tomcat.sh. Read that file for details on how to edit it.

55) Where do System.out and System.err go in Tomcat?

By default, they will go to the console where the tomcat is started. You have to modify the startup scripts to redirect them to appropriate files.

On the other hand, if you are running Tomcat as an NT Service, you can modify the conf/wrapper.properties file and set wrapper.stdout and wrapper.stderr properties to point to your log files.

56) I've written a servlet that was using JServ and now I upgrade to use Tomcat. What would I need to change?

The only thing you should have to change code-wise is whatever has been changed in the Servlet API between the API used in JServ and the api used it Tomcat... just a couple that come to mind are

encodeURL() is now encodeUrl() and encodeRedirectURL is now encodeRedirectUrl(). You can get the entire list by looking at the specs for the

Servlet APIs on the Java site. There is usually a section in the specs that is "What's New in the API".

57) Can I do Servlet chaining in Tomcat? If so, how?

Not yet.

The next version (2.3) of the Servlet Spec will allow something like Servlet chaining. It's called "filters."

58) Can I use a Generic SAX/DOM XML parser for Tomcat instead of parser.jar? I tried using Xerces and get class not found errors for sun.xxxx parser during Tomcat initialization.

Try to configure Xerces as customized SAX/DOM implementation by system properties. By default JAXP uses `com.sun.xml.parser.DocumentBuilderFactoryImpl` and `com.sun.xml.parser.SAXParserFactoryImpl` (this is why you get `ClassNotFoundException` if you miss `parser.jar` from the classpath).

Example:

```
System.setProperty("javax.xml.parsers.DocumentBuilderFactory",  
    "org.apache.xerces.jaxp.DocumentBuilderFactoryImpl");  
System.setProperty("javax.xml.parsers.SAXParserFactory",  
    "org.apache.xerces.jaxp.SAXParserFactoryImpl");
```

59) Under what circumstances will a Servlet be reloaded?

That depends on the Servlet container.

Most of the Servlet containers reload the Servlet only if it detects the code change in the Servlet, not in the referenced classes.

In Tomcat's `server.xml` deployment descriptor, if you have mentioned

```
<Context path="/myApp"  
    docBase="D:/myApp/webDev"  
    crossContext="true"  
    debug="0"  
    reloadable="true"  
    trusted="false" >  
</Context>
```

The reloadable = true makes the magic. Every time the Servlet container detects that the Servlet code is changed, it will call the destroy on the currently loaded Servlet and reload the new code.

But if the class that is referenced by the Servlet changes, then the Servlet will not get loaded. You will have to change the timestamp of the servlet or stop-start the server to have the new class in the container memory.

60) Why do I get the compiler error "package javax.servlet does not exist" or "package javax.servlet not found in import"?

Try this: Assume that you have tomcat location like:

c:\tomcat

at the command prompt type:

set classpath=c:\tomcat\lib\servlet.jar

61) How can you set the session timeout in tomcat higher?

Does anybody know how you can set the session time in tomcat higher? Now it is 30 minutes, but I want to set it higher.

You can specify the timeout of a session in the deployment descriptor of your web application (web.xml):

```
<web-app>
  <session-config>
    <session-timeout>60</session-timeout>
  </session-config>
```

```
...
</web-app>
```

The number within the session-timeout element must be expressed in minutes

62) Is JavaScript supported as a JSP scripting language under Tomcat?

In short, no.

Putting things very very simply, Tomcat is essentially a Java Servlet Compiler, taking the .jsp file you write and translating it to a java servlet that runs in the background.

Java, according to the Apache development group that wrote Tomcat (known as the Jakarta group) conforms solely to the JSP 1.2 / Servlet 1.1 specification by Sun Microsystems. When they wrote Tomcat, they only had in mind to use Java as the "scripting language". In fact, Tomcat is built with java, thereby heavily integrating the technology.

I hope you know this, but JavaScript and Java are two entirely different languages (some things are similar, but most are not). The two technologies are by no means interchangeable, and this intuitively applies to Tomcat. To the best of my knowledge the Apache Tomcat developmental teams have no current plans to integrate other scripting technologies into the software (for many reasons, of which I won't go into here for brevity).

However, you may include JavaScript within the .jsp file (not within jsp interpretable `<% %>` tags), and it will be output to the client browser for interpretation just like in an HTML page. The key here is to realize that it has no interaction/functionality with the server before it is sent to the client.

#### 63) Does TOMCAT provide JDBC Connection Pooling ?

No.

Tomcat (talking about version 3.x.x) is the reference implementation of the webapp part of J2EE (ie, no EJB). In this spec connection pooling is not contemplated.

#### 64) How do I include a file in a JSP page?

Yes, there are two different ways to do that:  
The first one is done using the 'include' directive:

```
<%@ include file="relativeURL" %>
```

In this case the given page will be statically included inside the the jsp page.

The second include is done using the jsp tag 'include':

```
<jsp:include page="{relativeURL | <%= expression %>}" flush="true" />
```

With this syntax, the page will be included statically or dinamically, depending on the page itself.



If the page to be included is dynamic, parameters can be passed to it using this syntax:

```
<jsp:include page="{relativeURL | <%= expression %>}" flush="true" >
  <jsp:param name="parameterName"
    value="{parameterValue | <%= expression %>}" />
</jsp:include>
```

Note: with Jsp 1.1 the "flush" parameter can only be true.

65) Where to save the Images, HTML, and other static data in Tomcat?

Images and HTML shall be put inside the context of your application:

- somewhere inside /jakarta-tomcat-3.2.1/webapps/your\_context/
- somewhere inside the war file of your application

war file must be put inside /jakarta-tomcat-3.2.1/webapps/

An example of tree of a web application (in a .war archive) can be:

```
war-file
|
|----META-INF
|  |---MANIFEST.MF
|
|---WEB-INF
|  |---web.xml
|  |---jsp
|  |  |---taglib (.tld)
|  |---classes
|  |---(.class/.properties)
|---jsp
|  |---(.jsp/.html/.shtml)
|
|---images
|  |---(.gif/.png/.jpeg)
|
|---binaries
|  |---(.zip/.tar/...)
```

check examples in jakarta-tomcat-3.2.1/webapps/

66) How do I find out Tomcat's home directory from inside a running servlet?

When Tomcat is executed (and you can check both tomcat.sh and tomcat.bat in the tomcat's bin directory) the property 'tomcat.home' is set using -D. This means that you should be able to access it using System.getProperty("tomcat.home").

67) How to read a file from the WAR or webapp? If I have a configuration file, say an XML file, stored inside the WAR, how do I open it from a servlet? I.e.,

```
InputStream in = new FileInputStream(????);
```

The WAR file is automatically opened by the servlet container during the startup... at least this is what the servlet specs are saying (and it is happening with both Resin and Tomcat). Once you reach the servlet just use ServletContext.getRealPath() to get the file.

For example if the file is in a 'template' dir under the root of the context, use

```
getRealPath("/template/file.xml");
```

68) How and where do I change the option for JVM heap memory in Tomcat?

If you look into the tomcat.sh or tomcat.bat under the bin directory of tomcat, you can see an environment variable called TOMCAT\_OPTS.

You can use it for passing all the JVM specific parameters, like the -X or -D.

You can set it directly in the file, or set it in the environment before calling it. For example in unix you can do:

```
export TOMCAT_OPTS="-Xms15M -Xmx15M"
tomcat.sh start
```

69) Where are the servlet source files for JSPs compiled in Tomcat?

[JSP is first translated into servlet then compiled. So where are those servlet .java or .class files for JSP examples in Tomcat?]

They are located in the work directory of the TOMCAT\_HOME, under a directory that is normally called localhost\_<port>%20<context\_name>.

The source for JSPs from the examples webapp should be something like localhost\_8080%20examples.

70) What is the difference between startup.bat, shutdown.bat, and tomcat.bat?

tomcat.bat takes a parameter, "start" or "stop".

Running "startup" is the same as running "tomcat start". Running "shutdown" is the same as running "tomcat stop".

71) How do I assign the user's role without using a login form?

Basically the concept is that you can use the standard http authentication schemes (i.e.: basic, digest). This means that you can authenticate a user among a database (or, for example, another source) without developing html login forms.

72 ) What's the difference between PoolTcpConnector and SimpleTcpConnector ?

The main difference is that the Pool connector will create a pool of simple connectors, allowing the server to answer to more requests at the same time, while a simple connector can answer only a response at a time.

The simple connector hasn't been designed to be used unless you know that you will process only a request at a time.

73) Do I really need to build a complete 'webapp' just to add a simple 'Helloworld' servlet to the server?

Not at all. You can put a single servlet anywhere on your hard disk but you will have to configure your path in server.xml and servlet alias in web.xml as usual.

Let me be a bit more descriptive.

If I have my HelloWorld servlet under c:/foo/hello directory, i will set my path in server.xml like

```
<Context path="/hello" docBase="c:/foo/hello"></context>
```

And then in web.xml :

```
<servlet-name>hello</servlet-name>
<servlet-class>HelloWorld</servlet-class>
```

that tomcat considers while starting and you can start your servlet.

74) Why does Tomcat's first servlet request take so long?

Sounds like it is the SecureRandom class initializing. Please disable this (set the value in server.xml to "java.util.Random") and see if it helps.

However, be warned that this method leaves your sessions open to attack. It would take a dedicated hacker, but if you're storing any sensitive data, this is a big no-no. So put it back to SecureRandom when you're done testing this.

75) How do I disable port 8080? I want Tomcat to run only through my Web server, not directly through Tomcat.

Go to the TOMCAT\_HOME directory and edit server.xml, find :

```
className="org.apache.tomcat.service.PoolTcpConnector"
```

and wrap it in a comment so it's not interpreted. That's it !

76) How do I prevent users from viewing the contents of my WEB-INF directory?

Servlet Engines (e.g. Resin or Tomcat) should not allow directory browsing of WEB-INF. Tomcat 3.2, for example, had that same issue and they have fixed it in Tomcat 3.2.1. This was happening when Tomcat was used standalone (not just as a container, but even as a web server).

If this problem happens with a specific version of a standalone Resin, you should consider to try the latest version (I'm running 1.2.5) and, eventually, send a bug to Caucho.

Consider that this issue should happen when using the container in standalone mode. When the container (Resin, Tomcat or another one) is used just for serving servlet & jsp behind a more reliable and complete web server, like Apache, the problem is on Apache, that is serving everything else. When you ask for WEB-INF, in fact, Apache doesn't even connect to Tomcat, there is no reason.

So, if this is your scenario, you should add lines like these:

```
<Directory /WEB-INF>  
AllowOverride None  
Deny From All  
</Directory>
```

Inside the virtual host or whatever you think is appropriate and, obviously, changing "/WEB-INF" with the appropriate context.

77) What is the password for the Tomcat admin webapp?

You have to modify tomcat-users.xml in your \$TOMCAT\_HOME/conf directory to add an administrative role as in following example:

```

<tomcat-users>
  <user name="scott" password="tiger" roles="admin" />
  <user name="role1" password="tomcat" roles="role1" />
  <user name="both" password="tomcat" roles="tomcat,role1" />
</tomcat-users>

```

Then you can login with scott/tiger.

78) Do I need to create my own web.xml file?

No, you don't. But it's normally better to.

Normally the servlet container knows how to execute servlets and what to call for executing jsp's, so you can survive without defining your own web.xml file.

There are many advantages of the web application descriptor, though. For example if you have a servlet that it's part of a package, you would be able to assign it a name and a mapping, so instead of calling it, for example, with /servlet/com.foo.servlets.MyServlet, you can make it /MyServlet. Or, you can use a servlet name that it's common and when you change the class that you are using, you don't have to change the code or the pages that are using it.

Plus there are other interesting features, like parameter passing, securities and so on.

But, again, to answer to your question: yes, you don't need to create a web.xml file.

79) Can the web application reside on a machine other than where Tomcat is running?

If by "web application" you mean the set of jsp's, servlets, classes, and all the other required files, I'm afraid not. [Tomcat must be able to read the files from a locally mounted file system.]

What you can do, though, is to have Tomcat running on one machine and the web server running on another, but I don't think that you can have the files of the application be read through the net without some sort of file server mounting solution like ntfs, samba or similar.

80) Can anyone tell me the advantages and disadvantages between Tomcat, JRun, and JServ for Apache Web Server?

Product	Pros	Cons
Tomcat	<ul style="list-style-type: none"> <li>• Free as in both beer <i>and</i> speech</li> <li>• Active open-source</li> </ul>	<ul style="list-style-type: none"> <li>• Not the fastest implementation on the planet</li> </ul>

	development effort <ul style="list-style-type: none"> <li>• Very current in terms of servlet API compliance</li> </ul>	<ul style="list-style-type: none"> <li>• You're on your own for support</li> </ul>
JRun	<ul style="list-style-type: none"> <li>• Free as in beer for developers</li> <li>• Relatively speedy</li> <li>• Supported by Allaire (Macromedia)</li> </ul>	<ul style="list-style-type: none"> <li>• Costs money for production use</li> <li>• Versions that I've used (pre-3.0) had <i>lots</i> of configuration headaches</li> <li>• Slightly behind the times with updates (compared to open source projects)</li> </ul>
Apache JServ	Free as in both beer <i>and</i> speech	<ul style="list-style-type: none"> <li>• No longer in active development (Tomcat inherited this effort)</li> <li>• Has some thread-leaking issues, in my experience</li> <li>• Only supports servlet API 2.0 (not later)</li> <li>• You're on your own for support</li> </ul>

81) Does Tomcat support JMS (Java Messaging Service)?

Tomcat is just a servlet container, not an EJB container nor an application server, so it does not contains any JMS basic support.

82) How do I load an applet from a servlet or JSP? Where do I place my applet class files?

An applet is executed on the client side (browser), not on the server side (servlet container). You need to place the applet class files in a location accessible from the browser, which means you have to treat them like normal files (like HTML or GIF files that you want the browser to load). Thus they need to go in the webapp directory tree, but not in the WEB-INF subdirectory.

It is best to think of the set of applet class files as entirely different from the set of servlet class files. They will be loaded by different Virtual Machines, and may even have different versions of classes. It is a simple matter to configure your build environment (Ant or make) to create copies of common class files in the two different classpath directories.

Since the concept of "current directory" is kind of fuzzy in the Servlet API, it is usually easiest to make a separate directory just for your applet class files, and use the optional CODEBASE parameter to the APPLET tag. Here is an architecture that works well:

```
myapp/WEB-INF/classes/MyServlet.class
myapp/WEB-INF/classes/Util.class
myapp/index.html
```

```
myapp/applets/MyApplet.class  
myapp/applets/Util.class
```

Then if your servlet generates a line like:

```
out.println("&lt;APPLET CODE='MyApplet.class' WIDTH=50 HEIGHT=50  
CODEBASE='/applets'&gt;");
```

The browser will be able to load MyApplet.class and Util.class from your /applets web directory. It will not be able to load anything from WEB-INF/classes, since those files are accessible only by the servlet engine.

Note that loading an applet from a page generated by a servlet is much the same as loading an image. It is also easier to use a common "images" directory in that case, so that images can be loaded from any page regardless of "current" directory.

83) Is it possible in Tomcat to allow running all servlets from a directory without configuring each one in web.xml ?

maybe you can just use the same logic that is used by the servlet, calling an invoker for the given directory. I never tested it, but it sound right:

```
<servlet>  
  <servlet-name>  
    invoker  
  </servlet-name>  
  <servlet-class>  
    org.apache.tomcat.servlets.InvokerServlet  
  </servlet-class>  
</servlet>  
<servlet-mapping>  
  <servlet-name>  
    invoker  
  </servlet-name>  
  <url-pattern>  
    /myDirectory/*  
  </url-pattern>  
</servlet-mapping>
```

84) Using jsp:forward on Tomcat, why do I get the error java.lang.IllegalArgumentException at javax.servlet.http.HttpUtils.parseName(HttpUtils.java:285) ?

Tomcat misinterprets jsp:param tag of jsp:forward if the name is represented as variable, i.e.

```
<jsp:param name="frame" value="bottom"/>
```

works well but

```
<jsp:param name="<%= var %>" value="bottom"/>
```

doesn't. The latter is what we had in our code hence the exception.

85) How do I restrict access from servlets to certain directories?

It's not that difficult. You can simply use the Java Security, provided you have Java 2.

For example :

```
grant {  
    permission java.io.FilePermission "-", "read,write";  
}
```

This only allows to read and write in the current directory and its sub-directories. Write this in your tomcat.policy file and then start tomcat like this :

```
/path/to/java/bin/java -Djava.security.manager  
-Djava.security.policy=/path/to/tomcat/conf/tomcat.policy  
-Dtomcat.home=/path/to/tomcat org.apache.tomcat.startup.Tomcat "$@" &  
Don't forget the necessary permissions for tomcat to work properly.
```

86) How do I determine a class's file location at runtime?

This method returns a URL to the class file in question. From that you can manipulate it to get what you want.

```
public java.net.URL locateClass(Class pClass){  
    // get the name of the class (e.g. my.package.MyClass)  
    String className = pClass.getName();  
  
    // create the resource path for this class  
    String resourcePath = "/" + className.replace('.', '/') + ".class";  
  
    // get the resource as a url (e.g. file:/C:/temp/my/package/MyClass.class)  
    java.net.URL classURL = pClass.getResource(resourcePath);  
  
    return classURL;  
}
```



87) Does Tomcat 4 have a GUI?

Accessing `http://localhost:8080/manager` directly will not do anything. You can give command to it like:

`http://localhost:8080/manager/list`

You will get a page (very simple) that show all the active web applications.

88) How do I set session timeouts of greater than 30 minutes?

Tomcat session keeps expiring in spite of entries in `web.xml` to the effect of

```
<session-config>
  <session-timeout>60</session-timeout>  <!-- 30 minutes -->
</session-config>
```

89) Why can't Tomcat find my Oracle JDBC drivers in `classes111.zip`?

TOMCAT 4.0.1 on NT4 throws the foll exception when i try to connect to Oracle DB from JSP.

`javax.servlet.ServletException : oracle.jdbc.driver.OracleDriver`

Root Cause : `java.lang.ClassNotFoundException: oracle.jdbc.driver.OracleDriver`

But, the Oracle JDBC driver ZIP file (`classes111.zip`) is avbl in the system classpath.

---

My problem was solved. I copied the Oracle Driver class file (`classes111.zip`) in `%TOMCAT_HOME%\lib` directory and renamed it to `classes111.jar`. It worked.

Now i am able to connect to Oracle DB from TOMCAT 4.01 via Oracle JDBC-Thin Driver.

90) Can I use \*.ear files in Tomcat?

No, Tomcat is a Web container, not an EJB container.

91) Can I place my classes somewhere other than inside WEB-INF?

Sun's specifications for Servlet define only the `WEB-INF/classes` and `WEB-INF/lib` directories in order to make the web application portable.

If you are not interested in portability, you can still put all your classes into the CLASSPATH environment variable.

The startup script for Tomcat 3.2.x should automatically add that classpath to the one used by Tomcat, while, with version 4.0.x and 3.3.x, you definitely need to make a small change in the startup script to add the CLASSPATH environment to the one that is generated by the script to execute Tomcat.

The only issue is that the classes/jars you've added will be available to all the web applications running under that instance.

92) I am attempting to configure Tomcat's WebDav application (<http://localhost:8080/webdav>) to allow everyone to view the directory. However I need authorized users to be able to edit the content of the directory.

Here some additional info regarding the problem.

First I set the property in the webapps/webdav/web.xml file to allow read and write access. By default this gives everyone access to change content.

```
<init-pram>
  <pram-name>readonly</pram-name>
  <pram-value>>false</pram-value>
</init-pram>
```

I tested this and it worked as expected--everyone was able to make changes to the content. Next I attempted to change the security settings for the directory so only certain users could make changes.

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>The Entire Web Application</web-resource-name>
    <url-pattern>/*</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <role-name>tomcat</role-name>
  </auth-constraint>
</security-constraint>
```

```
<login-config>
  <auth-method>BASIC</auth-method>
  <realm-name>Tomcat Supported Realm</realm-name>
</login-config>
```

```
<security-role>
  <description>
    An example role defined in "conf/tomcat-users.xml"
  </description>
  <role-name>tomcat</role-name>
```

</security-role>

This modified things so you need to login as a user defined as having a role of "tomcat" in conf/tomcat-users.xml in order to edit *or view* anything in the webdav directory.

I need it to allow anyone to access the file over the web, but only prompt for a password if they try to modify the file.

Trying to remove the <security-role> section doesn't work.

---

Basically whithin the <web-resource-colletion> section of the web.xml file you can define waht methods are protected. So you could protect the PUT command while leaving the GET command open to everyone.

I added the following and the behavior is what I was trying for:

```
<web-resource-collection>
....
<http-method>DELETE</http-method>
<http-method>POST</http-method>
<http-method>PUT</http-method>
<http-method>LOCK</http-method>
</web-resource-collection>
```

This prompts for authentication when a user tries to lock a file for editing as well as POSTing and PUTting. I'm not sure if this will have any effect on forms ability to function and send stuff to a jsp page. (For example: will the user be asked for a password if they fill out and submit a form.) This isn't really a problem because all my webdav content will be in a separate folder at this point.

Hopefully that is useful to some other people as well.

93) Where should I copy the class file of applet in Tomcat? Where should I copy respective HTML file?

HTML (and .JSP) files are placed under the webapps/ROOT directory under the installation directory (or other than ROOT if you've defined a different context). Applet .class files would then be placed relative to the location of the HTML file. You can also specify a specific codebase by providing that paramter to the HTML file.

94) What's the initial user name and password for Jakarta Tomcat 4's admin tool?

As far as I know, there is no user that is assigned the 'admin' role after an installation of tomcat. (For security reasons I assume).

You can assign this role to a username/password combination of your choice in the /conf/tomcat-users.xml file.

95) I put my JAR file inside WEB-INF/lib, but my classes are not getting found. However, when i unzipped classes into WEB-INF/classes/, it works. What's up?

Some old versions of Tomcat do not include WEB-INF/lib/\*.jar in WebApp classpath. You may handle it updating to a newer version of tomcat or (as you did) unpacking jars into WEB-INF/classes/.

96) Why request.getRemoteUser() returns NULL ?

As specified by the documentation, this method *returns the login of the user making this request, if the user has been authenticated, or null if the user has not been authenticated.*

The authentication, in this case, is the http authentication.

### Application Server

- 1) How can I get access to the properties of my running WebLogic server that have been set inside the weblogic.properties file? In particular I am after the document root of my server.

In WebLogic 5.1, to gain access to the weblogic properties, you can use the ConfigServicesDef interface which is accessible from the T3Client. The particular property that you are after is :

- weblogic.httpd.documentRoot

You may need to pre-pend the following properties to get the full path

- weblogic.system.home
- weblogic.system.name

The following is some sample code that should get you going

```
import weblogic.common.*;

public class SimpleT3Client {

    public static void main(String[] argv) throws Exception {
```

```

T3Client t3 = new T3Client("t3://localhost:7001");
t3.connect();
String prop = t3.services.config()
    .getProperty("weblogic.httpd.documentRoot");
t3.disconnect();
}
}

```

2) What is the difference between WebLogic Express server and WebLogic server?

WebLogic Express is a "cut-down" version of the WebLogic Server and contains the JDBC connection pool and Servlet functionality but amongst others, does not contain an EJB Container or transaction services.

3) What are the tradeoffs between using a single JRE versus running multiple, separate instances of the JVM for a complex application like an application server?

One of the major problems with JVM is garbage collection. When running small, short-lived applications there are not significant problems with Garbage Collection.

However, in the case of an Application Servers, which must service clients for long durations, memory usage is high and the amount of work required by the Garbage Collector is in turn high. This can lead to the following problems: Slow Garbage Collection or the Garbage Collector taking large amounts of time/resources to complete its' cycles.

So, for scalability, it is better to run multiple VMs - making each VM use only 100 - 200 megs of memory.

4) How do I configure my Application Server to show a different Error page upon an exception from a JSP or Servlet?

There should be one answer to this question that works for all 2.2 compliant servlet containers. The answer is to edit the web.xml deployment descriptor for your web app and add an error-page tag. The error-page tag tells the container the location of a page to return when an exception or error occurs. You can use the same mechanism to return customized error pages based either on a specific exception that is thrown, or based on the HTTP status code that is returned. The tag looks like this:

```

<error-page>
<exception-type>java.lang.NullPointerException</exception-type>
<location>myPage.html</location>
</error-page>

```

Or

```
<error-page>
<error-code>500</error-code>
<location>myPage.html</location>
</error-page>
```

- 5) When are stubs & skeletons created for my EJB components? is it at development time, deployment time or run time?

The time of creation of stubs and skeletons for EJB components varies from server to server.

The creation of stubs and skeletons for the popular servers is outlined below:

#### *Inprise Application Server 4.x*

The stubs and skeletons can be created during development time, for development and testing with the command line utility of with [JBuilder](#). The stubs and skeletons for the EJB can also be created during deployment using the IAS Console. There is also an option to create a Client Jar file. The Client Jar file contains the necessary stubs and interfaces for the EJB which a client requires to invoke the EJB.

#### *WebLogic Server 4.5.x, 5.x*

The stubs and skeletons are created at deployment time using the command line utility called weblogic.ejbc. In WebLogic 5.x, there is also a Deployer Tool available that will invoke the ejbc utility automatically.

#### *jBoss Server*

In jBoss, the stubs and skeletons are not required to be generated by the developer or deployer. The Server takes care of the creation and downloading of stubs.

- 6) What are the features/limitations of Container Managed Persistence with Weblogic 5.x. Are there any recommendations when using it?
- You can't specify storage of objects. For instance, Oracle 8i does support storage of objects, i.e Blobs and Clobs.
  - You are limited to the String objects and a few other primitive types. (For storage of arrays of values, you may need to find a work around. )
  - An Entity Bean can only map to one table, if more complex joins are required, you may require a more complex persistence manager( such as CocoBase or TopLink)
- 7) In WebLogic 4.51 and 5.x, what is the weblogic.system.executeThreadCount used for? What is its maximum and recommended value(s)?

The weblogic.system.executeThreadCount property is used to configure the number of execute threads inside the execute thread pool that service requests made to the WebLogic server. The maximum value for this parameter is

operating system dependent and recommended setting will vary upon your application behavior.

A general strategy for determining the optimal value for your system is to monitor your CPU utilization whilst running a test application.

If the WebLogic server is not responding to requests made from a client, *and* the CPU utilization is not at least 75% +, *then* you should increase the value and re-run the test.

- 8) Is there an example of writing a WebLogic CMP EJB that has a user-defined finder method with multiple parameters?

For Weblogic 5.1 the finder method is written as follows:

```
findByAccount(String str,String str1)
```

For the above finder method we write in the deployment descriptor:

```
<finder>
<method-name>findByAccount</method-name>
<method-params>
<method-param>java.lang.String </method-param>
<method-param>java.lang.String </method-param>
</method-params>

<finder-query>< ! [CDATA[(& (= name $0) (= address $1))]]> </finder-query>
</finder>
```

- 9) What is a weblogic.properties file? How can I create a new one?

The weblogic.properties file is used to configure the Weblogic server. It is used in versions 5.x and previous. In version 6.0, it has been deprecated and replaced with an xml file (config.xml).

The weblogic.properties file is a plain text file that follows the standard Java property file format - see this [FAQ for more details](#). All configuration information for the Weblogic server is placed inside this file, including:

- Deployments of EJB's and Servlets
- Definitions of JDBC Connection Pools and DataSources
- System configuration - Port and Address settings
- Security and Access Control List definitions

An example of a property from a weblogic.properties file is as follows:

```
weblogic.system.listenPort=7001
```

To create a new weblogic.properties file, simply create a new text file with the name weblogic.properties.

10) In WebLogic 5.x, how can I turn off logging to the console window?

You can do this by setting the following property in your [weblogic.properties](#) file

`weblogic.system.enableConsole=false`

11) What is a third party tool?

First party tools would be something you create yourself. Second party tools would be tools you buy from a vendor, like an IDE. Third party tools are tools for use with second party tools that aren't from the second party vendor. Think of them as add-ons to make second party tools more useful.

12) How do you deploy EJB components inside the iPlanet Application Server?

The iPlanet deployment tool, in my opinion, is the simplest way to deploy EJBs. However, for a more flexible and faster way using command line, the process is the following:

*Example assumes bean called Foo*

1. Compile your source files: `Foo.java`, `FooHome.java`,
2. 2. Generate the stubs using `ejbc.exe`. The format is: usage (typical): `EJBC <options> <home> <remote> <impl>`  
usage (RMIC mode): `EJBC <options> -rmic <remote>`  
options:
  - sl Compile as stateless session bean
  - sf Compile as stateful session bean
  - cmp Compile as CMP entity bean
  - iiop Generate additional CORBA classes
  - gs Generate java source files
  - d <dir> Output directory
  - help Show this message
  - rmic Generate RMIC code (see usage)
  - cp <classpath> Set classpath
  - javaccp <classpath> Prefix to javac classpath
3. Place your files and the compiled stubs in a directory along with the XML descriptors in a Jar compliant directory structure.
4. Generate a jar file (`Foo.jar`) that contains the descriptors, the stubs, your class files and the manifest.
5. Execute `ejbreg` as follows: `ejbreg Foo.jar`

This will register the bean. You will still need to make sure that the jar file is in iPlanet's classpath (`kregedit` for NT, `edit kjs` in Unix)

13) In WebLogic, How I can set my own property name and value in the *weblogic.properties* file and read that properties from a my code?



To do this, prepend your property with `java.system.property`.

and access your property using the `System` class. For example, if we have a property called "website" then we would add it to our property file as follows:

```
java.system.property.website=www.jguru.com
```

The code used to access this property would be as follows:

```
String s = System.getProperty("website");
```

14) In WebLogic, where do you find the classes for the `javax.naming` and `javax.sql` packages?

In WebLogic 5.1, these packages are located in

```
<weblogic dir> \lib\weblogicaux.jar
```

In WebLogic 6, they are located in

```
<weblogic dir>\lib\weblogic.jar
```

15) In WebLogic 5.1, what options are there for Session Persistence? How do they compare in relation to performance, and level of fail-over?

There are four different implementations of session persistence:

1. Memory (single-server, non-replicated)
2. File system persistence
3. JDBC persistence
4. In-Memory replication (across a cluster)

For customers who want the highest in servlet session persistence, JDBC-based persistence is the best choice.

For customers who want to sacrifice some amount of session persistence in favor of drastically better performance, in-memory replication is the appropriate choice.

JDBC-based persistence is noticeably slower than in-memory replication. In some cases, in-memory replication has outperformed JDBC-based persistence for servlet sessions by a factor of eight.

16) When calling an EJB from another EJB should they both be packaged in same JAR?. Are there any preferences for WebLogic or any other servers?

Definitely not. You can use any object published in JNDI by looking up it by name, obtain reference to it and narrowing this reference. As far as I know this can be done with any J2EE compatible server.

The only possible catch is to have the client classes available to the calling bean.

17) Which WebLogic 4.5x/5.x property is used to control the number of queued requests inside the WebLogic server?

The property that you are looking for is `weblogic.system.acceptBacklog`

18) When I try load \*.jsp files, I see the source code of each JSP file in my browser, what is wrong?

You can only load a JSP file through your webserver, if you open them by hand the browser will show the Java code inside the HTML file. By starting your Tomcat (or another JSP / Servlet engine) the server will compile the JSP and will send a 'normal' HTML file to your browser that will behave as you coded it in the Java code between the HTML tags.

If you did start the JSP / Servlet engine but you are still seeing the JSP code then the server mappings are wrong. The engine does not know that it should compile the JSP file that is requested and simply shows the complete file. In the Tomcat engine this is specified in the `server.xml` file in the `conf` directory. The `server.xml` should contain this:

```
<servlet>
  <servlet-name>
    jsp
  </servlet-name>
  <servlet-class>
    org.apache.jasper.runtime.JspServlet
  </servlet-class>
  <load-on-startup>
    -2147483646
  </load-on-startup>
</servlet>
```

This piece of XML tells the Tomcat server that it must use the `JspServlet` class when a JSP file is requested. The `JspServlet` then checks if the JSP file is compiled or not.

19) What can I do if my Servlet is receiving a `NullPointerException` when accessing init parameters via the `ServletConfig` object in WebLogic?

Make sure that your Servlet calls `super.init(config)` inside the `init()` method.

e.g.

```
public void init(ServletConfig config) {  
  
    ...  
    super.init(config);  
    ...  
}
```

20) How are Servlets and JSP's deployed inside WebLogic?

Place all your Servlet Class files in the following directory  
`weblogic/myserver/servletclasses/<package structure>` Add the following lines in  
your `weblogic.properties` file

```
weblogic.httpd.register.servlets=weblogic.servlet.ServletServlet
```

```
and weblogic.httpd.servlet.classpath=<drive name:>  
/weblogic/myserver/servletclasses/
```

Place all your JSP's in the following directory `weblogic/myserver/public_html`

21) In a WebLogic CMP EJB, how to obtain the maximum value in the specified  
database column? In WLQL is there something similar to `MAX()` function in SQL?

don't know how to write in WLQL. Another solution is to write the query using  
SQL inside the `weblogic-cmb-rdbms-jar.xml` file.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<finder-list>  
  <finder>  
    <method-name>findByOrgn</method-name>  
    <method-params>  
      <method-param>  
        <param-type>java.lang.String</param-type>  
        <param-name>orgnCode</param-name>  
      </method-param>  
    </method-params>  
    <finder-type>SQL</finder-type>  
    <finder-query>  
      <![CDATA[SELECT * FROM Asn_Goal_M WHERE ( orgn_code = #orgnCode  
and DELETE_FLAG='N') ]]>
```

```
</finder-query>
</finder>
```

22) When using JavaBeans from my JSP inside WebLogic, where should I place the JavaBean classes?

You should place these inside of a Web Application

If you are using an older version of WebLogic, you can place these classes on your Servlet Classpath.

23) What is the difference between an Application server and a Portal server?

A "portal server" is just an application server running a portal software or a portal application (Microsoft Sharepoint Portal Server is an example).

An application server is a system that provides the execution environment that is at the core of network computing or web-based architectures, providing a full set of services.

24) How do I use an EJB to handle simple authentication to an LDAP directory?

My recommendation would be to create a session bean that acquires an LDAP /JNDI context in the `setSession()` method. A subsequent call to `authenticate` would pass in a user and password. The following code shows how to authenticate the passed values:

```
env.put(Context.InitialContextFactory, "com.sun.jndi.ldap.LdapCtxFactory"); // I
added this line myself
env.put(Context.SECURITY_AUTHENTICATION, "simple");
env.put(Context.SECURITY_PRINCIPAL, "cn=S. User, ou=NewHires,
o=JNDITutorial");
env.put(Context.SECURITY_CREDENTIALS, "mysecret");
// Create the initial context
DirContext ctx = new InitialDirContext(env);
```

A failure in authentication will cause this code to throw a `javax.naming.AuthenticationException`

You may want to store the actual LDAP tree that contains the users you are authenticating so you can dynamically set it without recoding, so the actual code may look like this if you store the LDAP tree name in JNDI as "myUserLDAPTree" (`ou=NewHires, o=JNDITutorial`):

```
env.put(Context.InitialContextFactory, "com.sun.jndi.ldap.LdapCtxFactory"); // I
added this line myself
env.put(Context.SECURITY_AUTHENTICATION, "simple");
```

```
env.put(Context.SECURITY_PRINCIPAL, "cn=" + passedUserName + "," +
(String)new InitialContext().lookup("myUserLDAPTree"));
env.put(Context.SECURITY_CREDENTIALS, passedPassword);
// Create the initial context
DirContext ctx = new InitialDirContext(env);
```

The simply return a true, or a false if a javax.naming.AuthenticationException is thrown.

26) I want to set some user defined properties to be used in my application running in WebLogic appserver. In WebLogic 5.1, I was able to do this by setting them as System properties inside the weblogic.properties file like:

```
java.system.property.myproperty=myvalue
```

How do I achieve the same in WebLogic 6 ?

To be WebLogic 6 and J2EE compliant, you should use environment entries inside your xml descriptors for both EJB application archives and web application archives and use the JNDI environment naming context (ENC) to retrieve the values.

27) Differences in obtaining connection pools I'm trying to understand the difference between two ways to get a connection from a Weblogic 6.0 connection pool. What are the two ways and is either method preferable to the other? Or does it not matter?

The two different ways that we can get a connection from a pool are listed as follows(taken from the original posting from the author).

1. Have a connection pool manager with a getConnection method that uses the following code:

```
Connection conn = null;
Driver myDriver
=(Driver)Class.forName("weblogic.jdbc.pool.Driver").newInstance();
Properties props = new Properties();
props.put("connectionPoolID", "PoolName");
conn = myDriver.connect("jdbc:weblogic:pool", props);
```

2. Using JNDI and providing a URL (such as PROVIDER\_URL below) for the machine the connection pool is on. The following code could be used in a ConnectionPoolManager class too:

```

Context ctx = null;
Hashtable ht = new Hashtable();
ht.put(Context.INITIAL_CONTEXT_FACTORY,"weblogic.jndi.WLInitialContextFactory");
ht.put(Context.PROVIDER_URL,"t3://hostname:port");

try {
    ctx = new InitialContext(ht);
    javax.sql.DataSource ds =(javax.sql.DataSource)
ctx.lookup("myJtsDataSource");
    java.sql.Connection conn = ds.getConnection();
} catch (Exception ex) {}

```

The underlying result is the same. However, using a DataSource is the newer and much preferred method as the developer does not tie themselves to a particular driver or connection pool. There is also the added flexibility of changing connection pool properties without modifying the code.

27) Is there a simple example of how to use web application security in WebLogic?

Following is an example for WebLogic 6.x that protects all URLs that begin with /secure:

WEB-INF/web.xml - Define a constraint and a role

```

<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application
2.2//EN" "http://java.sun.com/j2ee/dtds/web-app_2_2.dtd">

```

```

<web-app>
  <security-constraint>
    <web-resource-collection>
      <web-resource-name>SecurePages</web-resource-name>
      <description>Security constraint for resources in the secure
directory</description>
      <url-pattern>/secure/*</url-pattern>
      <http-method>POST</http-method>
      <http-method>GET</http-method>
    </web-resource-collection>

    <auth-constraint>
      <description>only let the admin role access the pages
</description>
      <role-name>admin</role-name>
    </auth-constraint>

    <user-data-constraint>
      <description>SSL not required</description>

```

```

        <transport-guarantee>NONE</transport-guarantee>
    </user-data-constraint>
</security-constraint>

    <login-config>
        <auth-method>BASIC</auth-method>

    <security-role>
        <description>A role to access the secured pages</description>
        <role-name>admin</role-name>
    </security-role>
</web-app>

```

WEB-INF/weblogic.xml - Map the admin role to the system user in WebLogic.

```

<!DOCTYPE weblogic-web-app PUBLIC "-//BEA Systems, Inc.//DTD Web
Application 6.0//EN" "http://www.bea.com/servers/wls600/dtd/weblogic-web-
jar.dtd">

```

```

<weblogic-web-app>

    <security-role-assignment>
        <role-name>admin</role-name>
        <principal-name>system</principal-name>
    </security-role-assignment>

</weblogic-web-app>

```

28) What is a Multipool used for inside WebLogic?

You can think of a Multipool as a "pool of pools". WebLogic 6.x allows you to create a Multipool and have it mapped to a DataSource.

Multipools support two mutually exclusive algorithms:

1. Load Balancing - where requests are shared amongst the pools and,
2. High Availability - where the first pool is used until a fatal exception occurs, then control is passed to the second pool. Note that transaction information is not saved in the event of an error.

The client program is un-aware that a Mutlipool is being used, so there are no coding changes required, it is simply an administration change. Multipools work well with database failover products such as Oracle Parallel Server.

29) What are the valid values for the Session Descriptor inside the weblogic.xml file for WebLogic 6.1?

A list of valid parameter names for the Session Descriptor in webLogic 6.1 are as follows:

- CacheSize
- ConsoleMainAttribute
- CookieComment
- CookieDomain
- CookieMaxAgeSecs
- CookieName
- CookiePath
- CookiesEnabled
- IDLength
- InvalidationIntervalSecs
- JDBCConnectionTimeoutSecs
- PersistentStoreCookieName
- PersistentStoreDir
- PersistentStorePool
- PersistentStoreType
- SwapIntervalSecs
- TimeoutSecs
- TrackingEnabled
- URLRewritingEnabled

Each should be enclosed within a separate <session-pram> tag as follows:

```
<session-descriptor>
  <session-pram>
    <pram-name>URLRewritingEnabled</pram-name>
    <pram-value>True</pram-value>
  </session-pram>
  <session-pram>
    <pram-name>CookiesEnabled</pram-name>
    <pram-value>True</pram-value>
  </session-pram>
</session-descriptor>
```

### 30)How can I setup a Startup Class in WebSphere 3.5

IBM WebSphere provides the GUI where the option is there to set wether the server wants a particular Servlets is to be invoked upon start up. This will updated the XML as follows:

```
<servlet name="File Serving Enabler" action="update">
  <description>Auto-Generated - File Serving Servlet</description>
  com.ibm.servlet.engine.webapp.SimpleFileServlet
  <init-parameters/>
  <load-at-startup>true</load-at-startup>
  <debug-mode>>false</debug-mode>
```



```
<uri-paths>
  <uri value="/servlet/" />
</uri-paths>
<enabled>true</enabled>
</servlet>
```

31) Could you please help me??? What is the difference between Production mode and Development mode in WebLogic 6.1?

Placing the WebLogic server in Production mode will disable automatic deployment. This is the feature that allows you to automatically deploy an EJB, WAR or EAR file into the server by placing it into your configuration/applications directory.

This is turned off in production to reduce the overhead of continually polling the directory for updates.

32) Retrieving user name from inside each bean.

Inside an EJB you may retrieve the "Caller" name, that is the login id by invoking: `sessionContext.getCallerIdentity().getName()` where `sessionContext` is the instance of "SessionContext" (`setSessionContext`) passed to the Session Bean, or the instance of "EntityContext" (`setEntityContext`) passed to the Entity Bean.

33) How can I use Connection Pool?

*Additional info:*

I am using Oracle 9ias server. I have given max-connections to 50 in `data-sources.xml` file. And the class I am using is "oracle.jdbc.pool.OracleConnectionCacheImpl". I have also tried with `OracleConnectionPoolDataSource` class in `data-sources.xml`. But I feel that connection pool is not utilised, because in the middle of the retrieval, the server hangs as there will no be connections left for opening...

In entity beans, I have created connections in `setEntityContext` and releasing them in `unsetEntityContext`...

Do not get the connection in the `setEntityContext`. Get the connection only when you need it.

If you get the connection in `setEntityContext` with the pool of 50 connections and you retrieve 50 different entity beans each bean will hang on to the connection and you will hang.

So, get the connection when you need it and release the connection as soon as you have no need for it any longer.

There is no reason to get the connection in `setEntityContext`.

34) Are there any tools for porting EJB Applications from one Application Server to another?

In theory, you should not need any tools for that task.

If you have developed a fully compliant J2EE application, you will be able to deploy it on any Application Server that follows Sun J2EE Specifications. That is one of the biggest advantages of using standards and following the specifications.

*Maybe, then, you will probably have to spend some time in tuning and fixing the configuration of the specific server.*