## Historyof Java

Java history is interesting to know. The history of java starts from Green Team. Java team members (also known as Green Team), initiated a revolutionary task to develop a language for digital devices such as set-top boxes, televisions etc.

For the green team members, it was an advance concept at that time. But, it was suited for internet programming. Later, Java technology as incorporated by Netscape.

Currently, Java is used in internet programming, mobile devices, games, e-business solutions etc. There are given the major points that describes the history of java.

1) James Gosling, Mike Sheridan ,and Patrick Naught on initiated the Java language project in June 1991. The small team of sun engineers called Green Team.

2)Originally designed for small, embedded systems in electronic appliances like set-top boxes.

3)Firstly, it was called "Green talk" by James Gosling and file extension was .gt .

4)After that, it was called Oak and was developed as a part of the Green project.

Why Oak name for java language?

5) Why Oak? Oak is a symbol of strength and choosen as a national tree of many countries like U.S.A., France, Germany, Romania etc.

6) In1995, Oak was renamed as "Java" because it was already a trademark by Oak Technologies.

Why Java name for java language?

7) Why they choosed java name for java language? The team gathered to choose a new name. The suggested words were "dynamic", "revolutionary", "Silk", "jolt", "DNA" etc. They wanted something that reflected the essence of the technology: revolutionary, dynamic, lively, cool, unique, and easy to spell and fun to say.

According to James Gosling "Java was one of the top choices along with Silk". Since java was so unique, most of the team members preferred java.

8)Java is an island of Indonesia where first coffee was produced(called java coffee).

9)Notice that Java is just a name not an acronym.

10)Originally developed by James Gosling at Sun Microsystems (which isnow a subsidiary of Oracle Corporation) and released in 1995.

11) In1995, Time magazine called Java one of the Ten Best Products of 1995.

12)JDK 1.0 released in(January 23, 1996).

## Java Version History

There are many java versions that have been released. Current stable release of Java is Java SE 8.

JDK Alpha and Beta (1995)

JDK1.0 (23rd Jan, 1996)

JDK1.1 (19th Feb, 1997)

J2SE1.2 (8th Dec, 1998)

J2SE1.3 (8th May, 2000)

J2SE1.4 (6th Feb, 2002)

J2SE5.0 (30th Sep, 2004)

Java SE 6 (11th Dec, 2006)

Java SE 7 (28th July, 2011)

Java SE 8 (18th March, 2014)

## COREJAVA

**Abstraction**: Showing the essential and hiding the non-Essential is known as Abstraction.

**Encapsulation**: The Wrapping up of data and functions into a single unit is known as Encapsulation. Encapsulation is the term given to the process of hiding the implementation details of the object. Once an object is encapsulated, its implementation details are not immediately accessible any more. Instead they are packaged and are only indirectly accessed via the interface of the object.

**Inheritance** : is the Process by which the Object of one class acquires the properties of Object another Class. A reference variable of a Super Class can be assign to any Subclass derived from the Super class. Inheritance is the method of creating the new class based on already existing class, the new class derived is called Sub class which has all the features of existing class and its own, i.e. sub class.

Advantage : Reusability of code, accessibility of variables and methods of the Base class by the Derived class.

**Polymorphism** : The ability to take more than one form, it supports Method Overloading & Method Overriding.

**Method overloading** : When a method in a class having the same method name with different arguments(diff Parameters or Signatures) is said to be Method Overloading. This is Compile time Polymorphism.

Using one identifier to refer to multiple items in the same scope.

**Method Overriding** : When a method in a Class having same method name with same arguments is said to be Method overriding. This is Runtime Polymorphism.

Providing a different implementation of a method in a subclass of the class that originally defined the method.

1.In Overloading there is a relationship between the methods available in the same class, where as in overriding there is relationship between the Super class method and Sub class method.

2.Overloading does not block the Inheritance from the Super class, Where as in Overriding blocks Inheritance from the Super Class.

3.In Overloading separate methods share the same name, where as in Overriding Sub class method replaces the Super Class.

Overloading must have different method Signatures, Where as Overriding methods must have same Signatures.

**Dynamic dispatch** : is a mechanism by which a call to Overridden function is resolved at runtime rather than at Compile time, and this is how Java implements Run time Polymorphism.

**Dynamic Binding** : Means the code associated with the given procedure call is not known until the time of call the call at run time. (It is associated with Inheritance &Polymorphism).

**Bite code** : Is a optimized set of instructions designed to be executed by Java-runtime system, which is called the Java Virtual machine(JVM), i.e. in its standard form, the JVM is an Interpreter for byte code.

**JIT**-is a compiler for Byte code, The JIT-Complier is part of the JVM, it complies byte code into executable code in real time , piece-by-piece on demand basis.

**Final classes** : String, Integer, Color , Math

**Abstract class** : Generic Servlets, Number class

**Variable** : An item of data named by an identifier. Each variable has a type, such as inter Object, and a scope

**Class variable** : A data item associated with a particular class as a whole--not with particular instances of the class. Class variables are defined in class definitions. Also called a static field.

**Instance variable** : Any item of data that is associated with a particular object. Each instance of a class has its own copy of the instance variables defined in the class. Also called a field .

**Local variable** : A data item known within a block, but inaccessible to code out side the block. For example, any variable defined within a method is a local variable and can't be used outside the method.

**Class method** : A method that is invoked without reference to a particular object. Class methods affect the class as a whole, not a particular instance of the class. Also called a static method.

**Instance method** : Any method that is invoked with respect to an instance of a class. Also called simply a method.

**Interface** : Interfaces can be used to implement the Inheritance relationship between the non-related classes that do not belongs to the same hierarchy, i.e. any Class and any where in hierarchy. Using Interface, you can specify what a class must do but not how it does.

A class can implement more than one Interface.

An Interface can extend one or more interfaces, by using the keyword extends.

All the data members in the interface are public, static and Final by default.

An Interface method can have only Public, default and Abstract modifiers.

An Interface is loaded in memory only when it is needed for the first time.

A Class, which implements an Interface, needs to provide the implementation of all the methods in that Interface.

If the Implementation for all the methods declared in the Interface are not provided , the class itself has to declare abstract, other wise the Class will not compile.

If a class Implements two interface and both the Interface have identical method declaration, it is totally valid.

If a class implements two interfaces both have identical method name and argument list, but different return types, the code will not compile.

An Interface cannot be instantiated. Interface Are designed to support dynamic method resolution at run time.

An interface can not be native, static, synchronize, final, protected or private.

The Interface fields cannot be private or Protected.

A Transient variables and Volatile variables can not be members of Interface.

The extends keyword should not used after the Implements keyword, the Extends must always come before the Implements keyword.

A top level Interface can not be declared as static or final.

If an Interface species an exception list for a method, then the class implementing the interface need not declare the method with the exception list.

If an Interface cannot specify an exception list for a method, the class cannot throw an exception.

If an Interface does not specify the exception list for a method, the class can not throw any exception list.

The general form of Interface is

Access interface name {

return-typemethod-name1(parameter-list);

typefinal-varname1=value;

}

-----------------------

**Marker Interfaces** : Serializable , Closable, Remote, Event Listener,

Java.lang is the Package of all classes and is automatically imported into all Java Program

**Interfaces** : Clonable , Comparable, Runnable.

**Abstract Class** : Abstract classes can be used to implement the inheritance relationship between the classes that belongs same hierarchy.

Classes and methods can be declared as abstract.

Abstract class can extend only one Class.

If a Class is declared as abstract, no instance of that class can be created.

If a method is declared as abstract, the sub class gives the implementation of that class.

Even if a single method is declared as abstract in a Class , the class itself can be declared as abstract.

Abstract class has at least one abstract method and others may be concrete.

In abstract Class the keyword abstract must be used for method.

Abstract classes have subclasses.

Combination of modifiers Final and Abstract is illegal in java.

Abstract Class means - Which has more than one abstract method which doesn’t have method body but at least one of its methods need to be implemented in derived Class.

The general form of abstract class is :

Abstract type name (parameter list);

The Number class in the java.lang package represents the abstract concept of numbers. It makes sense to model numbers in a program, but it doesn't make sense to create a generic number object.

Difference Between Interfaces And Abstract class ?

All the methods declared in the Interface are Abstract, where as abstract class must have atleast one abstract method and others may be concrete.

In abstract class keyword abstract must be used for method, where as in Interface we need not use the keyword for methods.

Abstract class must have Sub class, where as Interface can’t have subclasses.

An abstract class can extend only one class, where as an Interface can extend more than one.

Public: The Variables and methods can be access any where and any package.

Protected: The Variables and methods can be access same Class, same Package &sub class.

Private: The variable and methods can be access in same class only.

Same class - Public,Protected, and Private

Same-package & subclass - Public, Protected

Same Package & non-sub classes - Public, Protected

Different package & Sub classes - Public, Protected

Different package & non- sub classes - Public

**Identifiers**: are the Variables that are declared under particular Data type.

**Literals** : are the values assigned to the Identifiers.

**Static**: access modifier. Sign a: Variable-Static int b; Method-static void meth(intx)

When a member is declared as Static, it can be accessed before any objects of its class are created and without reference to any object. Eg :main(),it must call before any object exit.

Static can be applied to Inner classes, Variables and Methods.

Local variables can’t be declared as static.

Astatic method can access only static Variables. and they canâ€™trefer to thisor superin any way.

Staticmethods canâ€™t be abstract.

Astatic method may be called without creating any instance of theclass.

Onlyone instance of static variable will exit any amount of classinstances.

Final: access modifier

Allthe Variables, methods and classes can be declared as Final.

Classesdeclared as final class canâ€™t be sub classed.

Methodâ€˜s declared as final canâ€™t be over ridden.

Ifa Variable is declared as final, the value contained in the Variablecanâ€™t be changed.

Staticfinal variable must be assigned in to a value in static initializedblock.

Transient: accessmodifier

Transientcan be applied only to classlevel variables.

Localvariables canâ€™t be declared as transient.

Duringserialization, Objectâ€™s transient variables are notserialized.

Transientvariables may not be final or static. But the complies allows thedeclaration and no compile time error is generated.

Volatile:access modifier

Volatileapplies to only variables.

Volatilecan applied to static variables.

Volatilecan not be applied to final variables.

Transientand volatilecan not come together.

Volatileis used in multi-processorenvironments.

Native: access modifier

Nativeapplies to only to methods.

Nativecan be applied to static methods also.

Nativemethods can not be abstract.

Nativemethods can throw exceptions.

Nativemethod is like an abstract method. The implementation of the abstractclass and native method exist some where else, other than the classin which the method is declared.

Synchronized: access modifier

Synchronizedkeyword can be applied to methods or partsof the methods only.

Synchronizekeyword is used to control the access to critical code inmulti-threaded programming.

Declarationof access specifier and access modifiers:

Class - Public, Abstract, Final

InnerClass - Public, Protected, Private, Final, Static,

Anonymous - Public, Protected, Private, Static

Variable - Public, Protected, Private, Final, Static, Transient, Volatile, Native

Method - Public, Protected, Private, Final, Abstract,Static, Native, Synchronized

Constructor - Public, Protected, Private

Free-floatingcode block - Static, Synchronized

Package: A Packageis a collection of Classes Interfaces that provides a high-levellayer of access protection and name space management.

Finalize()method:

Allthe objects have Finalize() method, this method is inherited from theObject class.

Finalize()is used to release the system resources other than memory(such asfile handles& network connecâ€™s.

Finalize() is used just before an object is destroyed and can be called priorto garbage collection.

Finalize()is called only once for an Object. If any exception is thrown in thefinalize() the object is still eligible for garbage collection.

Finalize()can be called explicitly. And can be overloaded, but only originalmethod will be called by Ga-collect.

Finalize() may only be invoked once by the Garbage Collector when the Objectis unreachable.

Thesignature finalize( ) : protectedvoid finalize() throws Throwable { }

Constructor() :

Aconstructor method is special kind of method that determines how anobject is initialized when created.

Constructorhas the same name as class name.

Constructordoes not have return type.

Constructorcannot be over ridden and can be over loaded.

Defaultconstructor is automatically generated by compiler if class does nothave once.

Ifexplicit constructor is there in the class the default constructor isnot generated.

Ifa sub class has a default constructor and super class has explicitconstructor the code will not compile.

Object:Object is a Super class for all the classes. The methods in Objectclass as follows.

Objectclone( ) final void notify( ) InthashCode( )

Booleanequals( ) final void notifyAll( )

Voidfinalize( ) String toString( )

FinalClass getClass( ) final void wait( )

Class: TheClass class is used to represent the classes and interfaces that areloaded by the JAVA Program.

Character: A class whose instances can hold a single character value. Thisclass also defines handy methods that can manipulate or inspectsingle-character data.

constructorsand methods provided by the Characterclass:

Character(char) : The Characterclass's only constructor, which creates a Characterobject containing the value provided by the argument. Once aCharacterobject has been created, the value it contains cannot be changed.

compareTo(Character):Aninstance method that compares the values held by two characterobjects.

equals(Object): Aninstance method that compares the value held by the current objectwith the value held by another.

toString(): An instance method that converts the object to a string.

charValue():An instance method that returns the value held by the characterobject as a primitive charvalue.

isUpperCase(char): A classmethod that determines whether a primitive charvalue is uppercase.

String: String is Immutableand String Is a final class.The Stringclass provides for strings whose value will not change.

Oneaccessor method that you can use with both strings and string buffersis the length()method, which returns the number of characters contained in thestring or the string buffer. The methods in String Class:-

toString() equals( ) indexOff() LowerCase( )

charAt() compareTo( ) lastIndexOff() UpperCase( )

getChars() subString( ) trim( )

getBytes() concat( ) valueOf( )

toCharArray() replace( )

ValueOf(): converts data from its internal formate into human readableformate.

StringBuffer: Is Mutable, TheStringBufferclass provides for strings that will be modified; you use stringbuffers when you know that the value of the character data willchange.

Inaddition to length,the StringBufferclass has a method called capacity,which returns the amount of space allocated for the string bufferrather than the amount of space used.

Themethods in StringBuffer Class:-

length() append( ) replace( ) charAt( ) and setCharAt( )

capacity() insert( ) substring( ) getChars( )

ensureCapacity() reverse( )

setLength() delete( )

WraperClasses :are the classes that allow primitive types to be accessed as Objects.

Theseclasses are similar to primitive data types but starting with capitalletter.

Number Byte Boolean

Double Short Character

Float Integer

Long

primitiveDatatypes in Java :

Accordingto Java in a Nutshell, 5th ed boolean, byte, char, short, longfloat, double, int.

Floatclass :The Float and Double provides the methods isInfinite( ) and isNaN( ).

isInfinite() : returns true if the value being tested is infinetly large orsmall.

isNaN() : returns true if the value being tested is not a number.

Characterclass: defines forDigit( ) digit( ) .

ForDigit() : returns the digit character associated with the value of num.

digit() : returns the integer value associated with the specifiedcharacter (which is presumably) according to the specified radix.

StringTokenizer: provide parsing process in which it identifies the delimitersprovided by the user, by default delimiters are spaces, tab, new lineetc., and separates them from the tokens. Tokens are those which areseparated by delimiters.

ObservableClass:Objects that subclass the Observable class maintain a list ofobservers. When an Observable object is updated it invokes theupdate( ) method of each of its observers to notify the observersthat it has changed state.

Observerinterface: is implemented by objects that observe Observable objects.

Instanceof() :is usedto check to see if an object can be cast into a specified type without throwing a cast class exception.

IsInstanceof() :determines if the specified Object is assignment-compatible with theobject represented by this class. This method is dynamic equivalentof the Java language instanceof operator. The method returns true ifthe specified Object argument is non-null and can be cast to thereference type represented by this Class object without raising aClassCastException. It returns false otherwise.

GarbageCollection: When an object is no longer referred to by any variable, javaautomatically reclaims memory used by that object. This is known asgarbage collection.

System.gc() method may be used to call itexplicitly and does not force the garbage collection but onlysuggests that the JVM may make an effort to do the GarbageCollection.

this(): can beused to invoke a constructor of the same class.

super() :can be used to invoke a super class constructor.

Innerclass :classes defined in other classes, including those defined in methodsare called inner classes. An inner class can have any accessibilityincluding private.

Anonymousclass :Anonymous class is a class defined inside a method without a name andis instantiated and declared in the same place and cannot haveexplicit constructors.

Whatis reflection API? How are they implemented

Reflectionpackage is used mainlyfor the purpose of getting the class name. byusing the getName method we can get name of the class for particularapplication. Reflection is a feature of the Java programminglanguage. It allows an executing Java program to examine or"introspect" upon itself, and manipulate internalproperties of the program.

Whatis heap in Java

JAVAis fully Object oriented language. It has two phases first one isCompilation phase and second one is interpratation phase. TheCompilation phase convert the java file to class file (byte code isonly readable format of JVM) than Intepratation phase interorate theclass file line by line and give the proper result.

main() : is the method where Java application Begins.

Stringargs[ ] : receives any command line argument during runtime.

System: is a predefined Class that provides access to the System.

Out: is output stream connected to console.

Println:displays the output.

Downcasting: is the casting from a general to a more specific type, i.e castingdown the hierarchy. Doing a cast from a base class to morespecific Class, the cast does;t convert the Object, just asserts itactually is a more specific extended Object.

Upcasting: byte can take Integer values.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_€€€€€€€€\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_€€\_\_€€€€€€

Exception

Exceptionhandling

Exceptioncan be generated by Java-runtime system or they can be manuallygenerated by code.

Error-Handlingbecomes a necessary while developing an application to account forexceptional situations that may occur during the program execution,such as

Runout of memory

Resourceallocation Error

Inabilityto find a file

Problemsin Network connectivity.

Ifthe Resource file is not present in the disk, you can use theException handling mechanism to handle such abrupt termination ofprogram.

Exceptionclass : is used for the exceptional conditions that are trapped by theprogram. An exception is an abnormal condition or error that occurduring the execution of the program.

Error: theerror class defines the conditions that do not occur under normalconditions.

Eg:Run out of memory, Stack overflow error.

Twotypes of exceptions:

1.CheckedExceptions: must be declare in the method declaration or caught in a catchblock.

Checkedexception must be handled at Compile Time. Environmental error thatcannot necessarly be detected by Testing, Eg: disk full, brockenSocket, Database unavailable etc.

2.Un-checkedExceptions:Run-time Exceptions and Error, doesâ€™t have to be declare.(but canbe caught).

Run-timeExceptions: programming errors that should be detectd in Testing ,

Arithmetic,Null pointer, ArrayIndexOutofBounds, ArrayStore, FilenotFound,NumberFormate, IO, OutofMemory.

Errors:Virtualmechine error â€“ class not found , out of memory, no such method ,illegal access to private field , etc.

JavaException handling can be managed by five keywords:

Try: The try block governs the statements that are enclosed within itand defines the scope of exception handler associated with it. Tryblock follows catch or finally or both.

Catch:This is a default exception handler. since the exception class is thebase class for all the exception class, this handler id capable ofcatching any type of exception.

Thecatch statement takes an Objectof exception class as a parameter,if an exception is thrown the statement in the catch block isexecuted. The catch block is restricted to the statements in theproceeding try block only.

Try{

// statements that may causeexception

}

catch(Exceptionobj)

{

}

Finally: when an exception is raised, the statement in the try block isignored,some times it is necessary to process certain statementsirrespectiveof wheather an exception is raisedor not,the finally block is used for this purpose.

Throw: The throw class is used to call exception explicitly. You maywant to throw an exception when the user enters a wrong login ID andpass word, you can use throw statement to do so.

Thethrow statement takes an single argument, which is an Object ofexception class.

Throw<throwableInstance>

Ifthe Object does not belong to a valid exception class the compilergives error.

Throws:The throws statement species the list of exception that has thrownby a method.

If a method is capable ofraising an exception that is does not handle, it must specify theexception has to be handle by the callingmethod, this is done by using the throw statement.

[<access specifier>][<access modifier>] <return type> <method name><arg-list> [<exception-list>]

Eg:public void accept password( ) throws illegalException

{

System.out.println(â€œIntruderâ€);

Throw newillegalAccesException;

}

MultiProgramming

Amultithreaded program contains two or more parts that can runconcurrently, Each part a program is called thread and each part thatdefines a separate path of execution.

Thusmultithreading is a specified from of multitasking .

Thereare two distinct types of multitasking .

Process: AProcess is , in essence , a program that is executing.

Process-based:is heavy weight- allows you run two or more programs concurrently.

Eg:you can use JAVA compiler at the same time you are using text editor.

Herea program is a small unit of code that can be dispatched by scheduler.

Thread-based:is Light weight- A Program can perform two or more taskssimultaneously.

Creatinga thread:

Eg:A text editor can formate at the same time you can print, as long asthese two tasks are being perform separate threads.

Thread:can be defined as single sequential flow of control with in aprogram.

SingleThread : Application can perform only one task at a time.

Multithreaded: Aprocess having more than one thread is said to be multithreaded.

Themultiple threads in the process run at the same time, performdifferent task and interact with each other.

DaemonThread :Is a low priority thread which runs immedeatly on the back grounddoing the Garbage Collection operation for the Java Run time System.

SetDaemon() â€“ is used to create DaemonThread.

Creatinga Thread :

1. By implementing the Runnable Interface.

Byextending the thread Class.

ThreadClass :Java.lang.Threadclass is used to construct and access the individualthreads in a multithreaded application.

Syntax:Public Class <class name> extends Thread { }

TheThread class define several methods .

Getname()â€“ obtain a thread name.

GetPriority()â€“ obtain thread priority.

Start() - start a thread by calling a Run( ).

Run() - Entry point for the thread.

Sleep() - suspend a thread for a period of time.

IsAlive() - Determine if a thread is still running.

Join() - wait for a thread to terminate.

RunableInterface : The Runnable interface consist of a Single method Run( ), which isexecuted when the thread is activated.

Whena program need it inherit from another class besides the threadClass, you need to implement the Runnable interface.

Syntax: public void <Class-name> extends <SuperClass-name>implements Runnable

Eg:public Class myapplet extends Japplet implements Runnable

{

// Implement the Class

}

\*Runnable interface is the most advantageous method to create threadsbecause we need not extend thread Class here.

NewThread :When an instance of a thread class is created, a thread enters thenew thread state. Thread newThread = new Thread(this);

You have toinvoke the Start( ) to start the thread. ie, newThread.Start( );

Runnable: when the Start( ) of the thread is invoked the thread enters intothe Runnable State.

NotRunnable : A thread is said to be not runnable state if it

ïƒ IsSlleping

ïƒ IsWaiting

ïƒ Isbeing blocked by another thread.

sleep(long t); wheret= no: of milliseconds for which the thread is inactive.

Thesleep( ) is a static method because it operates on the currentthread.

Dead: A threadcan either die natuarally or be killed.

-A thread dies a natural death when the loop in the Run( ) iscomplete.

-Assigning null to the thread Object kills the thread.

-If the loop in the Run( ) has a hundred iterations , the life of thethread is a hundred iterators of the loop.

IsAlive() : of thethread class is used to determine wheather a thread has been startedor stopped. If isAlive( ) returns true the thread is still runningotherwise running completed.

ThreadPriorities: are used by the thread scheduler to decide when each thread shouldne allowed to run.To set a thread priority, use te setpriority(), whichis a member of a thread.

final void setpriority(intlevel) - here level specifies the new priority seting for thecalling thread.

Thevalue level must be with in the range :-

MIN\_PRIORITY = 1

NORM\_PRIORITY= 5

MAX\_PRIORITY = 10

Youcan obtain the current priority setting by calling getpriority( ) ofthread.

finalint getpriority( )

Synchronization:

Two romore threads trying to access the same method at the same point oftime leads to synchronization. If that method is declared asSynchronized, only one thread can access it at a time. Another thread can accessthat method only if the first threadâ€™s task is completed.

Synchronizedstatement :Synchronized statements are similar to Synchronizedmethod.

ASynchronized statements can only be executed after a thread hasacquired a lock for the object or Class reffered in the Synchronizedstatements.

Thegeneral form is - Synchronized(object) {

//statements to be Synchronized

}

InterThread Communication : To Avoid pooling , Java includes an elegant interprocesscommunication mechanisim.

Wait() - tellsthe calling thread to give up the monitor and go to sleep until someother thread enters the same monitor & call notify( ).

notify( ) - wake up the first thread that called wait( ) on the same Object.

notifyall( ) â€“ wakeup all the threads that called wait( ) on the same Object.

The highestpriority thread aill run fast.

Serialization: The process of writing the state of Object to a byte stream totransfer over the network is known as Serialization.

Deserialization: and restored these Objects by deserialization.

Externalizable: is aninterface that extends Serializable interface and sends data intostreams in compressed format. It has two methods

WriteExternal(Objectoutputout)

ReadExternal(objectInputin)

I/OPackage Java.io.\*;

Thereare two classifications.

ByteStream - console input

CharacterStreamâ€“ File

1.ByteStream: Console Input

Read( ) - one character

Readline( ) â€“ one String

BufferReaderbr = new BufferReader(new InputStreamReader(System.in));

2.CharacterStream: File

FileInputStream - Storethe contents to the File.

FileOutStream - Get thecontents from File.

PrintWritepw = new printwriter(System.out.true);

Pw.println(â€œ â€œ);

Eg:-

Class myadd

{

publicstatic void main(String args[ ])

{

BufferReader br = newBufferReader(new InputStreamReader(System.in));

System.out.println(â€œEnterA no : â€œ);

int a =Integer.parseInt(br.Read( ));

System.out.println(â€œEnterB no : â€œ);

int b =Integer.parseInt(br.Read( ));

System.out.println(â€œTheAddition is : â€œ (a+b));

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_===\_€€€€€\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Collections

Collections :A collection allows a group of objects to be treated as a singleunit. collection define a set of core Interfaces as follows.

Collection Interface:The CI is the root of collection hierarchy and is used for commonfunctionality across all collections. There is no directimplementation of Collection Interface.

Set Interface:extends CollectionInterface. The Class Hash set implements Set Interface.

Is used to represent thegroup of uniqueelements.

Set stores elements in anunordered way but does notcontain duplicate elements.

Sorted set: extends SetInterface. The class TreeSetimplements Sorted set Interface.

It provides the extrafunctionality of keeping the elements sorted.

It represents the collectionconsisting of Unique,sorted elements in ascendingorder.

List :extendsCollection Interface. The classes ArrayList, Vector List & Linked List implements List Interface.

Represents the sequence ofnumbers in a fixed order.

But may contain duplicateelements.

Elements can be inserted orretrieved by their position in the List using Zero based index.

List stores elements in anordered way.

Map Interface:basicInterface.The classesHashMap & Hash Tableimplements Map interface.

Used to represent themapping of unique keys to values.

Byusing the key value we can retrive the values. Two basic operationsare get( ) & put( ) .

Sorted Map: extends MapInterface. The Class TreeMapimplements Sorted Map Interface.

Maintain the values of keyorder.

The entries are maintainedin ascending order.

HashSet: Implements Set Interface. HashSeths=new HashSet( );

The elements are not storedin sortedorder. hs.add(â€œmâ€);

TreeSet: Implements Sorted set Interface. TreeSetts=new TreeSet( );

The elements are stored insorted ascendingorder. ts.add(â€œHâ€);

Access and retrieval timesare quit fast, when storing a large amount of data.

Vector: Implements List Interface.

Vector implements dynamicarray. Vectorv = new vector( );

Vector is a growableobject. V1.addElement(newInteger(1));

Vector is Synchronized, it canâ€™t allow special charactersand nullvalues.

All vector starts withintial capacity, after it is reached next time if we want to storeobject in vector, the vector automatically allocates space for thatObject plus extra room for additional Objects.

ArrayList: Implements List Interface.

Array can dynamicallyincrease or decrease size. ArrayLista1=new ArrayList( );

Array List are ment forRandom ascessing. A1.add(â€œaâ€);

ArrayList are created with intial size, when the size is increased, thecollection is automatically enlarged. When an Objects are removed,the array may be shrunk.

LinkedList : Implements ListInterface.Inserting or removing elements in the middleof the array. LinkedListl1=new LinkedList( );

Linkedlist are meant for Sequentialaccessing. L1.add(â€œRâ€);

StoresObjects in a separate link.

MapClasses: Abstract Map;Hash Map ; Tree Map

HashMap : Implements MapInterface. Hashmap(), Hashmap(Map m), Hashmap(int capacity)

TheElements may not in Order.

HashMap is not synchronizedand permits null values

HashMap is not serialized. Hashmaphm = new HashMap( );

HashMap supports Iterators. hm.put(â€œHariâ€,newDouble(11.9));

HashTable : Implements MapInterface.

HashTable is synchronizedand does notpermit nullvalues.

HashTable is Serialized. Hashtableht = new Hashtable( );

Storeskey/value pairs in Hash Table. ht.put(â€œPrasadiâ€,newDouble(74.6));

A Hash Tablestores information by using a mechanism called hashing. In hashingthe informational content of a key is used to determine a uniquevalue, called its Hash Code. The Hash Code is then used as the indexat which the data associated with the key is stored. TheTransformation of the key into its Hash Code is performedautomatically- we never see the Hash Code. Also the code canâ€™tdirectly index into h c.

TreeMap : Implements Sorted SetInterface. TreeMaptm=new TreeMap( );

Theelements are stored in sortedascending order. tm.put(â€œPrasadâ€,new Double(74.6));

Usingkey valuewe can retrieve the data.

Providesan efficient means of storing key/valuepairs in sorted order andallows rapid retrivals.

Iterator:Each of collection class provided an iterator( ).

Byusing this iterator Object, we can access each element in thecollection â€“ one at a time.

Wecan remove() ; Hashnext( ) â€“ go next; if it returns false â€“endof list.

Iterarator Enumerator

Iteratoritr = a1.iterator( ); Enumerator vEnum =v.element( );

While(itr.hashNext()) System.out.println(â€œElements in Vector :â€);

{ while(vEnum.hasMoreElements( ) )

Object element = itr.next(); System.out.println(vEnum.nextElement( ) + â€œâ€œ);

System.out.println(element+ â€œ â€œ);

}

Collections

1.Introduction

2.Legacy Collections

1. The EnumerationInterface

2. Vector

3. Stack

4. Hashtable

5. Properties

3.Java 2 Collections

1. The Interfacesof the collections framework

2. Classes in thecollections framework

3. ArrayList &HashSet

4. TreeSet &Maps

Introduction :

Does your class need a wayto easily search through thousands of items quickly?

Does it need an orderedsequence of elements and the ability to rapidly insert and removeelements in the middle of the sequence?â€¢ Does it need an array likestructure with random-access ability that can grow at runtime?

The Enumeration Interface :

enumerate (obtain one at atime) the elements in a collection of objects.

specifies two methods:

boolean hasMoreElements() : Returnstrue when there are still more elements to extract, and false whenall of the elements have been enumerated.

Object nextElement() : Returnsthe next object in the enumeration as a generic Object reference.

VECTOR :

Vector implements dynamicarray. Vectorv = new vector( );

Vector is a growableobject. V1.addElement(newInteger(1));

Vector is Synchronized, it canâ€™t allow special charactersand nullvalues.

Vector is a variable-lengtharray of object references.

Vectors are created with aninitial size.

When this size is exceeded,the vector is automatically enlarged.

When objects are removed,the vector may be shrunk.

Constructors : Vector() : Defaultconstructor with initial size 10.

Vector(int size) : Vectorwhose initial capacity is specified by size.

Vector(int size,int incr) :Vectorwhose initialize capacity is specified by size and whose increment isspecified by incr.

Methods :

final voidaddElement(Object element) : Theobject specified by element is added to the vector.

final ObjectelementAt(int index) : Returnsthe element at the location specified by index.

final booleanremoveElement(Object element) : Removeselement from the vector

final boolean isEmpty() : Returnstrue if the vector is empty, false otherwise.

final int size() :Returnsthe number of elements currently in the vector.

final booleancontains(Object element) : Returnstrue if element is contained by the vector and false if it is not.

STACK :

Stack is a subclass ofVector that implements a standard last-in, first-out stack

Constructor: Stack() Creates anempty stack.

Methods :

Object push(Object item) :Pushes an itemonto the top of this stack.

Object pop() : Removesthe object at the top of this stack and returns that object as thevalue of this function. An EmptyStackException is thrown if it iscalled on empty stack.

boolean empty() : Testsif this stack is empty.

Object peek() : Looksat the object at the top of this stack without removing it from thestack.

int search(Object o) : Determineif an object exists on the stack and returns the number of pops thatwould be required to bring it to the top of the stack.

HashTable :

HashTable is synchronizedand does notpermit nullvalues.

HashTable is Serialized. Hashtableht = new Hashtable( );

Storeskey/value pairs in Hash Table. ht.put(â€œPrasadiâ€,newDouble(74.6));

Hashtable is a concreteimplementation of a Dictionary.

Dictionary is an abstractclass that represents a key/value storage repository.

A Hashtable instance can beused store arbitrary objects which are indexed by any other arbitraryobject.

A Hashtable storesinformation using a mechanism called hashing.

When using a Hashtable, youspecify an object that is used as a key and the value (data) that youwant linked to that key.

Constructors: Hashtable() Hashtable(int size)

Methods :

Object put(Objectkey,Object value) : Insertsa key and a value into the hashtable.

Object get(Object key) :Returnsthe object that contains the value associated with key.

boolean contains(Objectvalue) : Returnstrue if the given value is available in the hashtable. If not,returns false.

booleancontainsKey(Object key) : Returnstrue if the given key is available in the hashtable. If not, returnsfalse.

Enumeration elements() :Returns anenumeration of the values contained in the hashtable.

int size() : Returnsthe number of entries in the hashtable.

Properties

Properties is a subclass ofHashtable

Used to maintain lists ofvalues in which the key is a String and the value is also a String

Constructors

Properties()

Properties(PropertiespropDefault) : Createsan object that uses propDefault for its default value.

Methods :

String getProperty(Stringkey) : Returnsthe value associated with key.

Strng getProperty(Stringkey, String defaultProperty) : Returnsthe value associated with key. defaultProperty is returned if key isneither in the list nor in the default property list .

EnumerationpropertyNames() : Returnsan enumeration of the keys. This includes those keys found in thedefault property list.

Collection :

A collection allows a groupof objects to be treated as a single unit.

The Java collections libraryforms a framework for collection classes.

The CI is the root ofcollection hierarchy and is used for common functionality across allcollections.

There is no directimplementation of Collection Interface.

Two fundamental interfacesfor containers:

Collection

boolean add(Objectelement) : Insertselement into a collection

Set Interface:extends CollectionInterface. The Class Hash set implements Set Interface.

Is used to represent thegroup of uniqueelements.

Set stores elements in anunordered way but does notcontain duplicate elements.

identical to Collectioninterface, but doesnâ€™t accept duplicates.

Sorted set: extends SetInterface. The class TreeSetimplements Sorted set Interface.

It provides the extrafunctionality of keeping the elements sorted.

It represents the collectionconsisting of Unique,sorted elements in ascendingorder.

expose the comparison objectfor sorting.

List Interface:

ordered collection â€“Elements are added into a particular position.

Represents the sequence ofnumbers in a fixed order.

But may contain duplicateelements.

Elements can be inserted orretrieved by their position in the List using Zero based index.

List stores elements in anordered way.

Map Interface:Basic Interface.The classes HashMap & HashTableimplements Map interface.

Used to represent themapping of unique keys to values.

By using the key value wecan retrive the values.

Two basic operations areget( ) & put( ) .

boolean put(Object key,Object value) : Insertsgiven value into map with key

Object get(Object key) : Readsvalue for the given key.

TreeMap Class: ImplementsSorted Set Interface.

Theelements are stored in sortedascending order.

Usingkey valuewe can retrieve the data.

Provides an efficient meansof storing key/valuepairs insorted order and allows rapidretrivals.

TreeMaptm=new TreeMap( );

tm.put(â€œPrasadâ€,new Double(74.6));

TheClassesin Collections Framework

ArrayList

Similar to Vector: itencapsulates a dynamically reallocated Object[] array

Why use an ArrayList insteadof a Vector?

All methods of the Vectorclass are synchronized, It is safe to access a Vector object from two threads.

ArrayList methods are notsynchronized, use ArrayList in case of no synchronization

Use getand setmethods instead of elementAtand setElementAtmethods of vector

HashSet

Implements a set based on ahashtable

The default constructorconstructs a hashtable with 101 buckets and a load factor of 0.75

HashSet(intinitialCapacity)

HashSet(intinitialCapacity,float loadFactor)

loadFactor is a measure ofhow full the hashtable is allowed to get before its capacity isautomatically increased

Use Hashset if you donâ€™tcare about the ordering of the elements in the collection

TreeSet

Similar to hash set, withone added improvement

A tree set is a sortedcollection

Insert elements into thecollection in any order, when it is iterated, the values areautomatically presented in sorted order

Maps : Two implementations for maps:

HashMap

hashes the keys

TheElements may not in Order.

HashMap is not synchronizedand permits null values

HashMap is not serialized.

Hash Map supports Iterators.

TreeMap

uses a total ordering on thekeys to organize them in a search tree

The hash or comparisonfunction is applied onlyto the keys

The values associated withthe keys are not hashed or compared.

How are memory leakspossible in Java

If any object variable isstill pointing to some object which is of no use, then JVM will notgarbage collect that object and object will remain in memorycreating memory leak

What are the differencesbetween EJB and Java beans

the main difference is Ejbcomponenets are distributed which means develop once and runanywhere. java beans are not distributed. which means the beanscannot be shared .

What would happen if yousay this = null

this will give a compilationerror as follows

cannot assign value to finalvariable this

Will there be aperformance penalty if you make a method synchronized? If so, can youmake any design changes to improve the performance

yes.the performancewill be down if we use synchronization.

one can minimize the penaltyby including garbage collection algorithm, which reduces the cost ofcollecting large numbers of short- lived objects. and also by usingImproved thread synchronization for invoking the synchronizedmethods.the invoking will be faster.

How would you implement athread pool

public class ThreadPoolextends java.lang.Object implements ThreadPoolInt

This class is an genericimplementation of a thread pool, which takes the following input

a) Size of the pool to beconstructed

b) Name of the class whichimplements Runnable (which has a visible default constructor)

and constructs a thread poolwith active threads that are waiting for activation. once the threadshave finished processing they come back and wait once again in thepool.

This thread pool engine canbe locked i.e. if some internal operation is performed on the poolthen it is preferable that the thread engine be locked. Lockingensures that no new threads are issued by the engine. However, thecurrently executing threads are allowed to continue till they comeback to the passivePool

How does serializationwork

Its like FIFO method (firstin first out)

How does garbagecollection work

There are several basicstrategies for garbage collection: reference counting, mark-sweep,mark-compact, and copying. In addition, some algorithms can do theirjob incrementally (the entire heap need not be collected at once,resulting in shorter collection pauses), and some can run while theuser program runs (concurrent collectors). Others must perform anentire collection at once while the user program is suspended(so-called stop-the-world collectors). Finally, there are hybridcollectors, such as the generational collector employed by the 1.2and later JDKs, which use different collection algorithms ondifferent areas of the heap

How would you pass ajava integer by reference to another function

Passing by reference isimpossible in JAVA but Java support the object reference so.

Object is the only way topass the integer by refrence.

What is the sweep andpaint algorithm

The painting algorithm takesas input a source image and a list of brush sizes. sweep algo is thatit computes the arrangement of n lines in the plane ... a correctalgorithm,

Can a method be staticand synchronized

no a static mettod can't besynchronized

Do multiple inheritancein Java

Its not possible directly.That means this feature is not provided by Java, but it can beachieved with the help of Interface. By implementing more than oneinterface.

What is dataencapsulation? What does it buy you

The most common example Ican think of is a javabean. Encapsulation may be used by creating'get' and 'set' methods in a class which are used to access thefields of the object. Typically the fields are made private while theget and set methods are public.

Encapsulation can be used tovalidate the data that is to be stored, to do calculations on datathat is stored in a field or fields, or for use in introspection(often the case when using javabeans in Struts, for instance).

What is reflection API?How are they implemented

Reflection package is usedmainlyfor the purpose of getting the class name. by using the getNamemethod we can get name of the class for particular application .

Reflection is a feature ofthe Java programming language. It allows an executing Java program toexamine or "introspect" upon itself, and manipulateinternal properties of the program.

What are the primitivetypes in Java

According to Java in aNutshell, 5th ed

boolean, byte, char, short,long float, double, int

Is there a separate stackfor each thread in Java

No

What is heap in Java

JAVA is fully Objectoriented language. It has two phases first one is Compilation phaseand second one is interpratation phase. The Compilation phase convertthe java file to class file (byte code is only readable format ofJVM) than Intepratation phase interorate the class file line by lineand give the proper result.

In Java, how are objects/ values passed around

In Java Object are passed byreference and Primitive data is always pass by value

Do primitive types have aclass representation

Primitive data type has awrapper class to present.

Like for int - Integer ,for byte Byte, for long Long etc ...

How all can you freememory

With the help of finalize()method.

If a programmer really wantsto explicitly request a garbage collection at some point, System.gc()or Runtime.gc() can be invoked, which will fire off a garbagecollection at that time.

Does java do referencecounting

It is more likely that theJVMs you encounter in the real world will use a tracing algorithm intheir garbage-collected heaps

What does a static innerclass mean? How is it different from any other static member

A static inner class behaveslike any ``outer'' class. It may contain methods and fields.

It is not necessarily thecase that an instance of the outer class exists even when we havecreated an instance of the inner class. Similarly, instantiating theouter class does not create any instances of the inner class.

The methods of a staticinner class may access all the members (fields or methods) of theinner class but they can access only static members (fields ormethods) of the outer class. Thus, f can access the field x, but itcannot access the field y.

How do you declareconstant values in java

Using Final keyword we candeclare the constant values How all can you instantiate finalmembers Final member can be instantiate only at the time ofdeclaration. null

How is serializationimplemented in Java

A particular class has toimplement an Interface java.io.Serializable for implementingserialization. Whenyou have an object passed to a method and when the object isreassigned to a different one, then is the original reference lost No Reference is not lost. Java always passes the object by reference,now two references is pointing to the same object.

What are the differentkinds of exceptions? How do you catch a Runtime exception

There are 2 types ofexceptions.

1. Checked exception

2. Unchecked exception.

Checked exceptionis catched at the compile time while uncheckedexceptionis checked at run time.

1.CheckedExceptions : Environmental error that cannot necessarily be detectedby testing; e.g. disk full, broken socket, database unavailable, etc.

2. Unchecked exception.

Errors : Virtual machineerror: class not found, out of memory, no such method, illegal accessto private field, etc.

Runtime Exceptions:Programming errors that should be detected in testing: index out ofbounds, null pointer, illegal argument, etc.

Checked exceptions must behandled at compile time. Runtime exceptions do not need to be. Errorsoften cannot be

What are the differencesbetween JIT and HotSpot

The Hotspot VM is acollection of techniques, the most significant of which is called"adaptive optimization.

The original JVMsinterpreted bytecodes one at a time. Second-generation JVMs added aJIT compiler, which compiles each method to native code upon firstexecution, then executes the native code. Thereafter, whenever themethod is called, the native code is executed. The adaptiveoptimization technique used by Hotspot is a hybrid approach, one thatcombines bytecode interpretation and run-time compilation to nativecode.

Hotspot, unlike a regularJIT compiling VM, doesn't do "premature optimization"

What is a memoryfootprint? How can you specify the lower and upper limits of the RAMused by the JVM? What happens when the JVM needs more memory?

when JVM needs more memorythen it does the garbage collection, and sweeps all the memory whichis not being used.

What are thedisadvantages of reference counting in garbage collection?

An advantage of this schemeis that it can run in small chunks of time closely interwoven withthe execution of the program. This characteristic makes itparticularly suitable for real-time environments where the programcan't be interrupted for very long. A disadvantage of referencecounting is that it does not detect cycles. A cycle is two or moreobjects that refer to one another, for example, a parent object thathas a reference to its child object, which has a reference back toits parent. These objects will never have a reference count of zeroeven though they may be unreachable by the roots of the executingprogram. Another disadvantage is the overhead of incrementing anddecrementing the reference count each time. Because of thesedisadvantages, reference counting currently is out of favor.

Is it advisable to dependon finalize for all cleanups

The purpose of finalizationis to give an opportunity to an unreachable object to perform anyclean up before the object is garbage collected, and it is advisable.

can we declare multiplemain() methods in multiple classes. ie can we have each main methodin its class in our program?

\_\_\_\_\_\_\_\_\_\_\_€\_€\_\_\_\_\_€€€\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

JDBC

Howto Interact with DB?

Generallyevery DB vendor provides a User Interface through which we can easilyexecute SQL queryâ€™s and get the result (For example Oracle QueryManager for Oracle, and TOAD (www.quest.com) tool common to all thedatabases). And these tools will help DB developers to createdatabase. But as a programmer we want to interact with the DBdynamically to execute some SQL queries from our application (Anyapplication like C, C++, JAVA etc), and for this requirement DBvendors provide some Native Libraries (Vendor Specific) using this wecan interact with the DB i.e. If you want to execute some queries onOracle DB, oracle vendor provides an OCI (Oracle Call Interface)Libraries to perform the same.

AboutODBC

Whatis ODBC

ODBC(Open Database Connectivity) is an ISV (Independent software vendorproduct) composes of native API to connect to different databasesthrough via a single API called ODBC.

OpenDatabase Connectivity (ODBC) is an SQL oriented applicationprogramming interface developed by in collaboration with IBM and someother database vendors.

ODBCcomes with Microsoft products and with all databases on Windows OS.

Advantages

SingleAPI (Protocol) is used to interact with any DB

Switchingfrom one DB to another is easy

Doesnâ€™trequire any modifications in the Application when you want to shiftfrom one DB to other.

Whatfor JDBC?

Aswe have studied about ODBC and is advantages and came to know that itprovides a common API to interact with any DB which has an ODBCService Providerâ€™s Implementation written in Native API that can beused in your applications.

Ifan application wants to interact with the DB then the options whichhave been explained up to now in this book are:

UsingNative Libraries given by the DB vendor

UsingODBC API

Andwe have listed there Advantages and Disadvantages.

Butif the application is a JAVA application then the above given optionsare not recommended to be used due to the following reasons

NativeLibraries given by DB vendor

Applicationbecomes vendor dependent and

Theapplication has to use JNI to interact with Native Lib which maycause serious problem for Platform Independency in ourapplications.

Andthe second option given was using ODBC API which can solve the 1.aproblem but again this ODBC API is also a Native API, so we have touse JNI in our Java applications which lead to the 1.b describedproblem.

Andthe answer for these problems is JDBC (JavaDataBaseConnectivity)which provides a common Java API to interact with any DB.

Whatis JDBC

Asexplained above JDBC standards for JavaDataBaseConnectivity.It is a specification given by Sun Microsystems and standardsfollowed by X/Open SAG (SQL Access Group) CLI (Call Level Interface)to interact with the DB.

Javaprograming language methods. The JDBC API providesdatabase-independent connectivity between the JAVA Applications and awide range of tabular data bases. JDBC technology allows anapplication component provider to:

Performconnection and authentication to a database server

Managetransactions

MovesSQL statements to a database engine for preprocessing and execution

Executesstored procedures

Inspectsand modifies the results from SELECTstatements

JDBCAPI

JDBCAPI is divided into two parts

JDBCCore API

JDBCExtension or Optional API

JDBCCore API (java.sql package)

Thispart of API deals with the following futures

Establisha connection to a DB

GettingDB Details

GettingDriver Details

maintainingLocal Transaction

executingqueryâ€™s

gettingresultâ€™s (ResultSet)

preparingpre-compiled SQL queryâ€™s and executing

executingprocedures & functions

JDBCExt OR Optional API (javax.sql package)

Thispart of API deals with the following futures

ResourceObjects with Distributed Transaction Management support

ConnectionPooling.

Thesetwo parts of Specification are the part of J2SE and are inheritedinto J2EE i.e. this specification API can be used with all thecomponentâ€™s given under J2SE and J2EE.

In the above show archetecture diagram the JDBC Driver forms anabstraction layer between the JAVA Application and DB, and isimplemented by 3rdparty vendors or a DB Vendor. But whoever may be the vendor and whatever may be the DB we need not to worry will just us JDCB API to giveinstructions to JDBC Driver and then itâ€™s the responsibility ofJDBC Driver Provider to convert the JDBC Call to the DB SpecificCall.

Andthis 3rdparty vendor or DB vendor implemented Drivers are classified into4-Types namely

TypesOf Drivers :

Type-1(JDBC ODBC-Bridge Driver) JDBC-ODBCBridge Driver

Type-2(Java-Native API Driver) NativeAPI Partly JAVA Driver (Thick Driver)

Type-3(Java Net Protocol Driver) IntermediateDataBase Access Server

Type-4(Java Native Protocol driver) PureJAVA Driver (Thin driver)

Type-1: JDBC-ODBCBridge Driver:

SinceODBC is written in C-language using pointers, so JAVA doesâ€™tsupport pointers, a java program canâ€™t communate directly with theDataBase. The JDBC-ODBC bridge drivertransulates JDBC API calls toODBC API calls.

Thistype of Driver is designed to convert the JDBC request call to ODBCcall and ODBC response call to JDBC call.

TheJDBC uses this interface in order to communicate with the database,so neither the database nor the middle tier need to be Javacompliant. However ODBC binary code must be installed on each clientmachine that uses this driver. This bridge driver uses a configureddata source.

Advantages

Simpleto use because ODBC drivers comes with DB installation/Microsoftfront/back office product installation

JDBCODBC Drivers comes with JDK software

Disadvantages

Morenumber of layers between the application and DB. And more number ofAPI conversions leads to the downfall of the performance.

Slowerthan type-2 driver

Whereto use?

Thistype of drivers are generaly used at the development time to testyour applicationâ€™s.

Becauseof the disadvantages listed above it is not used at production time.But if we are not available with any other type of driverimplementations for a DB then we are forced to use this type ofdriver (for example MicrosoftAccess).

Examplesof this type of drivers

JdbcOdbcDriverfrom sun

Sunâ€™sJdbcOdbcDriver is one of type-1 drivers and comes along with sunj2sdk (JDK).

Settingenvironment to use this driver

Software

ODBClibraries has to be installed.

classpath

Noadditional classpath settings are required apart from the runtimejar (c:\j2sdk1.4\jre\lib\rt.jar) which is defaultly configured.

Path

Noadditional path configuration is required.

Howto use this driver

Driverclass name Ã sun.jdbc.odbc.JdbcOdbcDriver

DriverURL Ã dbc:odbc:<DSN>

here<DSN> (DataSourceName)is an ODBC datasource name which is used by ODBC driver to locateone of the ODBC Service Provider implementation API which canin-turn connect to DB.

Stepsto create <DSN>

runâ€˜DataSources (ODBC)â€™from Control Panal\Administrative Tools\

(for Windows 2000server/2000 professional/XP)

run â€˜ODBCData Sourcesâ€™from Control Panel\

2.click on Addbutton available on the above displayed screen. this opens a newwindow titled â€˜CreateNew Data Sourceâ€™which displays all the available DBâ€™s lable DBâ€™s ODBC driverscurrently installed on your system.

3.Select the suitable driver and click on Finish

4.Give the required info to the driver (like username, service id etc)

Type-2: NativeAPI Partly JAVA Driver (Thick Driver):

JDBCDatabase calls are translated into Vendor-specific API calls. Thedatabase will process the request and send the results back throughAPI to JDBC Driver â€“ this will translate the results to the JDBCstandard and return them to the Java application.

TheVendor specific language API must be installed on every client thatruns the JAVA application.

Thisdriver converts the JDBC call given by the Java application to a DBspecific native call (i.e. to C or C++) using JNI (JavaNativeInterface).

Advantages:Fasterthan the other types of drivers due to native library participationin socket programing.

Disadvantage: DBspecifiic native client library has to be installed in the clientmachine.

Preferabllywork in local network environment because network service name mustbe configured in client system

Whereto use?

Thistype of drivers are suitable to be used in server side applications.

Notrecommended to use with the applications using two tier model (i.e.client and database layerâ€™s) because in this type of model clientused to interact with DB using the driver and in such a situation theclient system sould have the DB native library.

Examplesof this type of drivers

1.OCI 8 (OracleCallInterface)for Oracle implemented by Oracle Corporation.

Settingenvironment to use this driver

Software:Oracle client software has to be installed in client machine

classpathÃ %ORACLE\_HOME%\ora81\jdbc\lib\classes111.zip

pathÃ %ORACLE\_HOME%\ora81\bin

Howto use this driver

Driverclass nameÃ oracle.jdbc.driver.OracleDriver

DriverURLÃ jdbc:oracle:oci8:@TNSName

Note:TNS Names ofOracle is available in Oracle installed folder%ORACLE\_HOME%\Ora81\network\admin\tnsnames.ora

2.Weblogic Jdriver for Oracle implementedby BEA Weblogic:

Settingenvironment to use this driver

Oracleclient software has to be installed in client machine

weblogicoicdllâ€™s has to be set in the path

classpathÃ d:\bea\weblogic700\server\lib\weblogic.jar

pathÃ %ORACLE\_HOME%\ora81\bin;

d:\bea\weblogic700\server\bin\<subfolder><sub folder> is

oci817\_8if youare using Oracle 8.1.x

oci901\_8forOracle 9.0.x

oci920\_8forOracle 9.2.x

Howto use this driver

Driverclass nameÃ weblogic.jdbc.oci.Driver

DriverURLÃ jdbc:weblogic:oracle:HostName

Type-3IntermediateDataBase Access Server :

Type-3Driver uses an Intermediate(middleware) database driver that has theability to connect multiple JAVA clients to multiple databaseservers.

Clientconnect to the Databse server via an Intermediate server component(such as listener) that acts as a gateway for multple databaseservers.

Beaweblogic includesType-3 Driver.

Architecture:

Thistype of drivers responsibility is to convert JDBC call to Netprotocol (Middleware listener dependent) format and redirect theclient request to Middleware Listener and middleware listener inturnuses type-1, type-2 or type-4 driver to interact with DB.

Advantages:

Itallows the flexibility on the architecture of the application.

Inabsence of DB vendor supplied driver we can use this driver

Suitablefor Applet clients to connect DB, because it uses Java libraries forcommunication between client and server.

Disadvantages:

Fromclient to server communication this driver uses Java libraries, butfrom server to DB connectivity this driver uses native libraries,hence number of API conversion and layer of interactions increasesto perform operations that leads to performance deficit.

Thirdparty vendor dependent and this driver may not provide suitabledriver for all DBs

Whereto use?

Suitablefor Applets when connecting to databases

Examplesof this type of drivers:

1.IDS Server (Intersolv)driver available for most of the Databases

Settingenvironment to use this driver

Software:IDS software required to be downloaded from the following URL

[http://www.idssoftware.com/idsserver.html-> Export Evaluation ]

classpathÃ C:\IDSServer\classes\jdk14drv.jar

pathÃ

Howto use this driver

Driverclass nameÃ ids.sql.IDSDriver

DriverURLÃ jdbc:ids://localhost:12/conn?dsn='IDSExamples'

Note:DSN Name mustbe created in ServerDSN

Type-4 PureJAVA Driver (Thin driver):

Type-4Driver translates JDBC-API calls to direct network calls usingvendor specific networking protocols by making direct serverconnections with the database.

Thistype of driver converts the JDBC call to a DB defined nativeprotocol.

Advantage

Type-4driver are simple to deploy since there is No client nativelibraries required to be installed in client machine

Comeswith most of the Databases

Disadvantages:

Slowerin execution compared with other JDBC Driver due to Java librariesare used in socket communication with the DB

Whereto use?

Thistype of drivers are sutable to be used with server sideapplications, client side application and Java Applets also.

Examplesof this type of drivers

1)Thin driver forOracle implemented by Oracle Corporation

Settingenvironment to use this driver

classpathÃ %ORACLE\_HOME%\ora81\jdbc\lib\classes111.zip

Howto use this driver

Driverclass nameÃ oracle.jdbc.driver.OracleDriver

DriverURLÃ jdbc:oracle:thin:@HostName:<portno>:<SID>

<portno> Ã 1521

<SID>-> ORCL

2)MySQL Jconnector forMySQL database

Settingenvironment to use this driver

classpathÃ C:\mysql\mysql-connector-java-3.0.8-stable\mysql-connector-java-3.0.8-stable-bin.jar

Howto use this driver

Driverclass nameÃ com.mysql.jdbc.Driver

DriverURLÃ jdbc:mysql:///test

Chapter3 [JDBC Core API]

Inthis chapter we are going to discuss about 3 versions of JDBC: JDBC1.0, 2.0 and 3.0

Q)How JDBC API is common to all the Databases and also to all drivers?

A)Fine! The answer is JDBC API uses Factory Method and Abstract FactoryDesign pattern implementations to make API common to all theDatabases and Drivers. In fact most of the classes available in JDBCAPI are interfaces, where Driver vendors must provide implementationfor the above said interfaces.

Q)Then how JDBC developer can remember or find out the syntaxes ofvendor specific classes?

A)No! developer need not have to find out the syntaxes of vendorspecific implementations why because DriverManager is one named classavailable in JDBC API into which if you register Driver class name,URL, user and password, DriverManager class in-turn brings us oneConnection object.

Q)Why most of the classes given in JDBC API are interfaces?

A)Why abstract class and abstract methods are?

Abstractclass forces all sub classes to implement common methods whicheverare required implementations. Only abstract method and class can dothis job. Thatâ€™sâ€™ why most part of the JDBC API is a formation ofinterfaces.

JDBCAPI comes in 2 packages

java.sql.\*

javax.sql.\*

Firstof all I want to discuss briefly about all the list of interfaces andclasses available in java.sql. package

Interfacesindex

Driver

EveryJDBC Driver vendor must one sub class of this class for initialestablishment of Connections. DriverManager class need to be firstregistered with this class before accepting URL and other informationfor getting DB connection.

Methodindex

Connectionconnect(String url, Properties info)

This method takes URLargument and user name & password info as Properties object

booleanacceptURL(String url)

This method returns boolean value true ifthe given URL is correct, false if any wrong in URL

booleanjdbcComplaint()

JDBCcompliance requires full support for the JDBC API and full supportfor SQL 92 Entry Level. It is expected that JDBC compliant driverswill be available for all the major commercial databases.

Connection

Connectionis class in-turn holds the TCP/IP connection with DB. Functionsavailable in this class are used to manage connection live-ness aslong as JDBC application wants to connect with DB. The period for howlong the connection exists is called as Session. This class alsoprovides functions to execute various SQL statements on the DB. Forinstance the operations for DB are mainly divided into 3 types

DDL(create, alter, and drop)

DML(insert, select, update and delete)

DCL(commit, rollback) and also

callfunction\_name (or) call procedure\_name

MethodIndex

StatementcreateStatement()

PreparedStatementprepareStatement(String preSqlOperation)

CallableStatementprepareCall(String callToProc())

Statement

Statementclass is the super class in its hierarchy. Provides basic functionsto execute query (select) and non-related (create, alter, drop,insert, update, delete) query operations.

MethodIndex

intexecuteUpdate(String sql)

Thisfunction accepts non-query based SQL operations; the return value inttells that how many number of rows effected/updated by the given SQLoperation.

ResultSetexecuteQuery(String sql)

Thisfunction accepts SQL statement SELECT and returns java buffer objectwhich contains temporary instance of SQL structure maintaining allthe records retrieved from the DB. This object exists as long as DBconnection exist.

booleanexecute()

Thisfunction accepts all SQL operations including SELECT statement also.

PreparedStatement

PreparedStatementclass is sub classing from Statement class. While connection classprepareStatement function is creating one new instance this class,function takes one String argument that contains basic syntax of SQLoperation represented with â€œ?â€ for IN parameter representation.In the further stages of the JDBC program, programmer uses setXXX(intindex, datatype identifier) to pass values into IN parameter andrequests exdcute()/ exuecteUpdate() call.

MethodIndex

setInt(intindex, int value) â€“ similar functions are provided for all otherprimitive parameters

setString(intindex, String value)

setObject(intindex, Object value)

setBinaryStream(intindex, InputStream is, int length)

CallableStatement

ResultSet ResultSetMetaData DatabaseMetaData

BLOB CLOB REF

SavePoint Struct

SQLInput SQLOutput SQLData

Classdiagram required here

//TypeI DriverTest,java

packagecom.digitalbook.j2ee.jdbc;

importjava.sql.\*;

publicclass TypeIDriverTest

{

Connectioncon;

Statementstmt;

ResultSetrs;

publicTypeIDriverTest ()

{

try{

//Load driver class into default ClassLoader

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

//Obtain a connection with the loaded driver

con=DriverManager.getConnection("jdbc:odbc:digitalbook","scott","tiger");

URLString - ("<protocol>:<subprotocol>:<subname>"," ", " " ); }

//create a statement

st=con.createStatement();

//executeSQL query

rs=st.executeQuery ("select ename,sal from emp");

System.out.println("Name Salary");

System.out.println("--------------------------------");

while(rs.next())

{

System.out.println(rs.getString(1)+" "+rs.getString(2));

}

rs.close();

stmt.close();

con.close();

}

catch(Exceptione)

{

e.printStackTrace();

}

}

publicstatic void main (String args[])

{

TypeIDriverTestdemo=new TypeIDriverTest ();

}

}

//TypeIIDriverTest,java

packagecom.digitalbook.j2ee.jdbc;

importjava.sql.\*;

publicclass TypeIIDriverTest

{

Connectioncon;

Statementstmt;

ResultSetrs;

publicTypeIIDriverTest ()

{

try{

//Load driver class into default ClassLoader

Class.forName("oracle.jdbc.driver.OracleDriver");

//Obtain a connection with the loaded driver

con=DriverManager.getConnection("jdbc:oracle:oci8:@digital","scott","tiger");

//create a statement

st=con.createStatement();

//executeSQL query

rs=st.executeQuery ("select ename,sal from emp");

System.out.println("Name Salary");

System.out.println("--------------------------------");

while(rs.next())

{

System.out.println(rs.getString(1)+" "+rs.getString(2));

}

rs.close();

stmt.close();

con.close();

}

catch(Exceptione)

{

e.printStackTrace();

}

}

publicstatic void main (String args[])

{

TypeIIDriverTestdemo=new TypeIIDriverTest ();

}

}

Chapter9 : [javax.sql package]

Thispackage supplements the java.sql package and is included as a part ofJDK 1.4 version. This package mainly provides following features:

DataSourceinterface was introduced in substitution to DriverManager class forgetting connection objects.

ConnectionPooling

DistributedTX management

RowSets

Applicationscan directly use DataSource and RowSet API but connection pooling andDistributed TX management APIs are used internally by the middle-tierinfrastructure.

DataSource

DataSourceis an interface. Driver vendor will provide implementation for thisinterface (That means in case JDBC Driver Type II driver Oraclevendor for Oracle DB, Intersolv in case of IDSServer). This object isused to obtain connections into any type of JDBC program. ThoughDriverManager class is ideal for getting DB connection object, thisclass provides some extra features over DriverManager class:

Applicationswill obtain DB connection objects through via this factory class

DataSourceobject will be registered into JNDI, hence any application connectedin the network can obtain this object by requesting JNDI API,DataSource class is having one method called getConnection() geivesone Connection object

Applicationdo not need to hard code a driver class

Changescan be made to a data source properties, which means that it is notnecessary to make changes in application code when something aboutthe data source or driver changes

Connectionpooling and Distributed transactions are available through only theconnection obtained from this object. Connection obtained throughDriverManager class do not have this capability

DataSourceinterface is implemented by driver vendor. There are 3 types ofimplementations available:

BasicImplementation- Producesa standard connection object.

ConnectionPooling Implementation- Producesa connection object that automatically participates in connectionpooling. This implementation works with a middle-tier connectionpooling manager.

Distributedtransaction implementation-Produces a connection object that may be used for distributedtransactions and almost always participates in connection pooling.This implementation works with a middle-tier transaction manager andalmost always with a connection pool manager.

Adriver that is accessed via a DataSource object does not registeritself with the DriverManager. Rather, a DataSource object isretrieved though a lookup operation and then used to create aConnection object. With a basic implementation, the connectionobtained through a DataSource object is identical to a connectionobtained through the DriverManager facility.

MethodIndex

ConnectiongetConnection() â€“ This function returns Connection object ondemand of this method.

ConnectiongetConnection(String user, String pass) â€“ This function returnsConnection object on demand of this method by passing username andpassword.

Subclasses of this interface are

TypeIII Driver â€“ IDSServer â€“ Intersolv â€“ ids.jdbc.IDSDataSource

TypeIII Driver â€“ WebLogic â€“ BEA â€“ weblogic.jdbc.jta.DataSource â€“XA Support

ConnectionPooling

Connectionsmade via a DataSource object that is implemented to work with amiddle tier connection pool manager will participate in connectionpooling. This can improve the performance dramatically becausecreating a new connection is very expensive.

ConnectionPool provides following features:

Substantialimprovement in the performance of DB application can be accomplishedby pre-caching the DB connection objects

CPMsupplied DB connections are remote enable

CPMsupplied DB connections are cluster aware

CPMsupplied DB connections supports DTM (distributed TXs)

CPMsupplied DB connections are not actual DB Connection objects, inturn they are remote object, hence even though client closes DBconnection using con.close() the actual connection may not be closedinstead RMI connection between client to CPM are closed

CPMsupplied DB connection objects are serializable, hence client fromany where in the network can access DB connections

Theclasses and interfaces used for connection pooling are:

ConnectionPoolDataSource

PooledConnection

ConnectionEvent

ConnectionEventListener

ConnectionPool Manager resided on middle tier system uses these classes andinterfaces behind the scenes. When the ConnectionPooledDataSourceobject is called on to create PooledConnection object, the connectionpool manager will register as a ConnectionEventListener object withthe new PooledConnection object. When the connection is closed orthere is an error, the connection pool manager (being listener) getsa notification that includes a ConnectionEvent object.

DistributedTransactions

Aswith pooled connections, connections made via data source object thatis implemented to work with the middle tier infrastructure mayparticipate in distributed transactions. This gives an applicationthe ability to involve data sources on multiple servers in a singletransaction.

Theclasses and interfaces used for distributed transactions are:

XADataSource

XAConnection

Theseinterfaces are used by transaction manager; an application does notuse them directly.

TheXAConnection interface is derived from the PooledConnectioninterface, so what applies to a pooled connection also applies to aconnection that is part of distributed transaction. A transactionmanager in the middle tier handles everything transparently. The onlychange in application code is that an application cannot do anythingthat would interfere with the transaction managerâ€™s handling of thetransaction. Specifically application cannot call the methodsConnection.commit or Connection.rollback and it cannot set theconnection to be in auto-commit mode.

Anapplication does not need to do anything special to participate in adistributed transaction. It simply creates connections to the datasources it wants to use via the DataSource.getConnection method, justas it normally does. The transaction manager manages the transactionbehind the scenes. The XADataSource interface creates XAConnectionobjects, and each XAConnection object creates an XAResource objectthat the transaction manager uses to manage the connection.

Rowsets

TheRowSetinterface works with various other classes and interfaces behind thescenes. These can be grouped into three categories.

EventNotification

RowSetListener

ARowSetobject is a JavaBeansTMcomponent because it has properties and participates in theJavaBeans event notification mechanism. The RowSetListenerinterface is implemented by a component that wants to be notifiedabout events that occur to a particular RowSetobject. Such a component registers itself as a listener with arowset via the RowSet.addRowSetListenermethod.

When theRowSetobject changes one of its rows, changes all of it rows, or movesits cursor, it also notifies each listener that is registered withit. The listener reacts by carrying out its implementation of thenotification method called on it.

RowSetEvent

Aspart of its internal notification process, a RowSetobject creates an instance of RowSetEventand passes it to the listener. The listener can use thisRowSetEventobject to find out which rowset had the event.

Metadata

RowSetMetaData

Thisinterface, derived from the ResultSetMetaDatainterface, provides information about the columns in a RowSetobject. An application can use RowSetMetaDatamethods to find out how many columns the rowset contains and whatkind of data each column can contain.

TheRowSetMetaDatainterface provides methods for setting the information about columns,but an application would not normally use these methods. When anapplication calls the RowSetmethod execute,the RowSetobject will contain a new set of rows, and its RowSetMetaDataobject will have been internally updated to contain information aboutthe new columns.

TheReader/Writer Facility

ARowSetobject that implements the RowSetInternalinterface can call on the RowSetReaderobject associated with it to populate itself with data. It can alsocall on the RowSetWriterobject associated with it to write any changes to its rows back tothe data source from which it originally got the rows. A rowset thatremains connected to its data source does not need to use a readerand writer because it can simply operate on the data source directly.

RowSetInternal

Byimplementing the RowSetInternalinterface, a RowSetobject gets access to its internal state and is able to call on itsreader and writer. A rowset keeps track of the values in its currentrows and of the values that immediately preceded the current ones,referred to as the originalvalues. A rowset also keeps track of (1) the parameters that havebeen set for its command and (2) the connection that was passed toit, if any. A rowset uses the RowSetInternalmethods behind the scenes to get access to this information. Anapplication does not normally invoke these methods directly.

RowSetReader

Adisconnected RowSetobject that has implemented the RowSetInternalinterface can call on its reader (the RowSetReaderobject associated with it) to populate it with data. When anapplication calls the RowSet.executemethod, that method calls on the rowset's reader to do much of thework. Implementations can vary widely, but generally a reader makes aconnection to the data source, reads data from the data source andpopulates the rowset with it, and closes the connection. A reader mayalso update the RowSetMetaDataobject for its rowset. The rowset's internal state is also updated,either by the reader or directly by the method RowSet.execute.

RowSetWriter

Adisconnected RowSetobject that has implemented the RowSetInternalinterface can call on its writer (the RowSetWriterobject associated with it) to write changes back to the underlyingdata source. Implementations may vary widely, but generally, a writerwill do the following:

Makea connection to the data source

Checkto see whether there is a conflict, that is, whether a value thathas been changed in the rowset has also been changed in the datasource

Writethe new values to the data source if there is no conflict

Closethe connection

The RowSetinterface may be implemented in any number of ways, and anyone maywrite an implementation. Developers are encouraged to use theirimaginations in coming up with new ways to use rowsets.

TypeIII Driver â€“ WebLogic â€“ BEA â€“weblogic.jdbc.common.internal.ConnectionPool

TypeIII Driver â€“ WebLogic â€“ BEA â€“weblogic.jdbc.connector.internal.ConnectionPool

TypeII & IV driver â€“ Oracle DB - Oracle â€“

JDBC:

Thereare three types of statements in JDBC

Createstatement: Is used to execute single SQL statements.

Preparedstatement:Is used for executing parameterized quaries. Is used to runpre-compiled SEQL Statement.

Callablestatement:Is used to execute stored procedures.

StoredProcedures:Is a group of SQL statements that perform a logical unit and performsa particular task.

Areused to encapsulate a set operations or queries t execute on data.

execute() â€“ returns Boolean value

executeupdate() â€“ returns resultset Object

executeupdate() â€“ returns integer value

Loadingthe Driver:

Class.forName(â€œsun.jdbc.odbc.JdbcOdbcDriverâ€);

Conn=DriverManager.getConnection(â€œjdbc:odbc:dsnâ€,â€œusernameâ€, â€œpasswordâ€);

(ORACLE Driver )

Class.forName(â€œOracle.jdbc.driver.OracleDriverâ€);

Conn=DriverManager.getConnection(â€œjdbc:oracle:thin:@192.168.1.105:1521:dbnâ€,â€œusernameâ€, â€œpasswordâ€);

Database connection:

Publicstatic void main(String args[]);

Connectioncon;

Statementst;

Resultsetrs;

try{ // Getting all rows from Table

Clas.forName(â€œsun.jdbc.odbc.jdbcodbcâ€);

Conn=DriverManager.getConnction(â€œjdbc.odbc.dsnâ€,â€œusernameâ€ , â€passwordâ€);

st= con.createstatement( );

rs= st.executestatement(â€œSELECT \* FROM mytableâ€);

while(rs.next());

{

Strings= rs.getString(1); or rs.setString(â€œCOL\_Aâ€);

inti = rs. getInt(2);

Floatf = rs.getfloat(3);

Process(s,i,f);

}

catch(SQLExceptione)

{}

//Gettingparticular rows from Table

st= con.createstatement( );

rs= st.executequery(â€œSELECT \* FROM mytable WHERE COL A = â€œPrasadâ€);

while(rs.next());

{

Strings = rs.getString(1);

Inti = rs.getint(2);

Floatf = rs.getfloat(3);

Process(s,i,f);

}

Catch(SQLExceptione); { }

//updatinga row from table.

try{

st= con.createstatement( );

intnumupdated = st.executeupdate(â€œUPDATE mytable SET COL\_A = â€œprasadâ€ WHERE COL\_B=â€746â€);

rs= st.executeupdate();

conn.close(); }

catch(SQLExceptione);{ }

//Receiving rows from table

try {

st= con.preparedstatement( );

rs= st.execurtestatement(â€œSELECT \* FROM mytable SET COL\_A=?â€™);

intcolunm=1;

rs.setString(colunm,â€hariâ€);

rs= st.executeQuery( );

//updaterwo from table

st= con.createstatement( );

intnumupdated = st.executeupdate(â€œUPDATE mytable SET COL\_A =? WHERECOL\_B=?â€);

intcolumn=1;

rs.setString(colunm,â€Prasadâ€);

intcolumn=2;

rs.setString(column,â€746â€);

intnumupdated = st.executeupdate( );

} catch(SqlException e); { }

//callablestatement

try{

cst= con.preparecall(â€œ{call add1(??,??)}â€);

cst.setint(1,a);

cst.setint(2,b);

cst.registerOurPrameter(1,Types.INTEGER);

cst.executeQuery();

System.out.println(â€œrs.getString()â€); }

ConnectionPool with webLogic server :

Youcan connect the database in your app using :

Class.forName(â€œweblogic.jdbc.oci.Driverâ€).newInstance();

Java.sql.Connectionconn = Driver.connect(â€œjdbc:weblogic:Oracle:dbnâ€, â€usernameâ€,â€œpasswordâ€);

(Or )

java.util.Propertiesprop = new java.util.Properties( );

prop.put(â€œuserâ€,â€œhariâ€);

prop.put(â€œpasswordâ€,â€prasadâ€);

java.sql.Driverd =(java.sql.Driver)Class.forName(â€œweblogic.jdbc.oci.Driverâ€).newInstance();

java.sql.Connectionconn = d.connect(â€œjdbc:weblogic:Oracle:dbnâ€, prop);

publicstatic void main(String args[]) throws Exception

{

java.sql.Connectioncon=null;

java.sql.satementst =null;

try{

contextctx=null;

Hashtableht = new Hashtable( );

ht.put(Context.INTIAL\_CONTEXT\_FACTORY,â€weblogic:jndi:WLInitialContextFACTORYâ€);

ht.put(Context\_PROVIDER\_URL,â€t3://Localhost:7001â€);

//geta context from JNDI lookup

ctx= newIntialContext( ):

java.sql.Datasourseds =(java.sql.DataSource)ctx.lookup(â€œOraclegbJNDIâ€);

con=ds.getConnection( );

System.out.Println(â€œMakingConnectionâ€¦â€¦â€);

st= conn.createstatement( );

}

finally{

try{

if(stmt!=null)

stmt.close();

if(stmt!=null)

con.close(); }

Whatis a transaction

transactionis collection of logical operation that perform a task

Transactionshould ACID properties.

Afor Automicity

Cfor Consistency

Ifor Isolation

Dfor Durability.

Atransaction can be termed as any operation such as storing,retrieving, updating or deleting records in the table that hits thedatabase.

Whatis the purpose of setAutoCommit( )

Itis set as

ConnectionObject.setAutoComit();

afterany updates through the program cannot be effected to the database.Wehave commit the transctions .For this puprpose we can set AutoCommitflag to Connection Object.

Whatare the three statements in JDBC & differences between them

whichis used to run simple sql statements like select and update

2.PrepareStatment is used to run Pre compiled sql.

3.CallableStatement is used to execute the stored procedures.

Whatis stored procedure. How do you create stored procedure ?

Storedprocedures is a group of SQL statements that performs a logical unitand performs a particular task.

Storedprocedures are used to encapsulate a set of operations or queries toexecute on data.

StoredProcedure is a stored program in database, PL/SQL program is a StoredProcedure. Stored Procedures can be called from java byCallableStatement

Aprecompiled collection of SQL statements stored under a name andprocessed as a unit.

Storedprocedures can:

1.Acceptinput parameters and return multiple values in the form of outputparameters to the calling procedure or batch.

2.Containprogramming statements that perform operations in the database,including calling other procedures.

3.Returna status value to a calling procedure or batch to indicate success orfailure (and the reason for failure).

Whatare batch updates?

BatchUpdate facility allows multiple update operations to be submitted toa database for processing at once. Using batch updates will improvethe performance.

Whatis the difference between Resultset and Rowset

ARowSet is a disconnected, serializable version of a JDBC ResultSet.

TheRowSet is different than other JDBC interfaces in that you can writea RowSet to be vendor neutral. A third party could write a RowSetimplementation that could be used with any JDBC-compliant database.The standard implementation supplied by Sun uses a ResultSet to readthe rows from a database and then stores those rows as Row objects ina Vector inside the RowSet. In fact, a RowSet implementation could bewritten to get its data from any source. The only requirement is thatthe RowSet acts as if it was a ResultSet. Of course, there is noreason that a vendor couldn't write a RowSet implementation that isvendor specific.

Thestandard implementations have been designed to provide a fairly goodrange of functionality. Theimplementations provided are:

CachedRowSetImpl- This is the implementation of the RowSet that is closest to thedefinition of RowSet functionality that we discussed earlier. Thereare two ways to load this RowSet. The execute ( ) method will loadthe RowSet using a Connection object. The populate( ) method willload the RowSet from a previously loaded ResultSet.

WebRowSetImpl- This is very similar to the CachedRowSetImpl (it is a child class)but it also includes methods for converting the rows into an XMLdocument and loading the RowSet with an XML document. The XMLdocument can come from any Stream or Reader/Writer object. This couldbe especially useful for Web Services.

JdbcRowSetImpl- This is a different style of implementation that is probably lessuseful in normal circumstances. The purpose of this RowSet is to makea ResultSet look like a JavaBean. It is not serializable and it mustmaintain a connection to the database.

Theremaining two implementations are used with the first threeimplementations:

FilteredRowSetImpl- This is used to filter data from an existing RowSet. The filterwill skip records that don't match the criteria specified in thefilter when a next() is used on the RowSet.

JoinRowSetImpl- This is used to simulate a SQL join command between two or moreRowSet objects.

Whatare the steps for connecting to the database using JDBC

UsingDriverManager:

1.Load the driver class using class.forName(driverclass) andclass.forName() loads the driver class and passes the control toDriverManager class

2.DriverManager.getConnection() creates the connection to the databse

UsingDataSource.

DataSourceis used instead of DriverManager in Distributed Environment with thehelp of JNDI.

1.Use JNDI to lookup the DataSource from Naming service server.

DataSource.getConnectionmethod will return Connection object to the database

Whatis Connection Pooling ?

Connectionpooling is a cache of data base connections that is maintained inmemory , so that the connections may be reuse.

Connectionpooling is a place where a set of connections are kept and are usedby the different programers with out creating conncections to thedatabase(it means there is a ready made connection available for theprogrammers where he can use). After using the connection he can sendback that connection to the connection pool. Number of connections inconnection pool may vary.

Howdo you implement Connection Pooling

ConnectionPooling can be implemented by the following way.

\*A javax.sql.ConnectionPoolDataSource interface that serves as aresource manager connection factory for pooled java.sql.Connectionobjects. Each database vendors provide the implementation for thatinterface.

Forexample, the oracle vendors implementation is as follows:

oracle.jdbc.pool.oracleConnectionPoolDataSourceClass.

Ajavax.sql.PooledConnection interface encapsulates the physicalconnection for the database. Again, the vendor provides theimplementation.

WhatClass.forName( ) method will do

Class.forName()is used to load the Driver class which is used to connect theapplication with Database. Here Driver class is a Java class providedby Database vendor.

Whatis the difference between JDBC 1.0 and JDBC 2.0

TheJDBC 2.0 API includes many new features in the java.sql package aswell as the new Standard Extension package, javax.sql. This new JDBCAPI moves Java applications into the world of heavy-duty databasecomputing. New features in the java.sql package include support forSQL3 data types, scrollable result sets, programmatic updates, andbatch updates. The new JDBC Standard Extension API, an integral partof Enterprise JavaBeans (EJB) technology, allows you to writedistributed transactions that use connection pooling, and it alsomakes it possible to connect to virtually any tabular data source,including files and spread sheets.

TheJDBC 2.0 API includes many new features like

Scrollableresult sets

Batchupdates

ConnectionPooling

Distributedtransactions

setautocomit ( )

Whatis JDBC?

JDBCis a layer of abstraction that allows users to choose betweendatabases. It allows you to change to a different database engine andto write to a single API. JDBC allows you to write databaseapplications in Java without having to concern yourself with theunderlying details of a particular database.

Whatare the two major components of JDBC?

Oneimplementation interface for database manufacturers, the otherimplementation interface for application and applet writers.

Whatis JDBC Driver interface?

TheJDBC Driver interface provides vendor-specific implementations of theabstract classes provided by the JDBC API. Each vendors driver mustprovide implementations of thejava.sql.Connection,Statement,PreparedStatement, CallableStatement,ResultSet and Driver.

Whatare the common tasks of JDBC?

Createan instance of a JDBC driver or load JDBC drivers throughjdbc.drivers

Registera driver

Specifya database

Opena database connection

Submita query

Receiveresults

Whatpackages are used by JDBC?

Thereare 8 packages: java.sql.Driver, Connection,Statement,PreparedStatement, CallableStatement, ResultSet, ResultSetMetaData,DatabaseMetaData.

Whatare the flow statements of JDBC?

AURL string-->getConnection-->DriverManager-->Driver-->Connection-->Statement-->executeQuery-->ResultSet.

1).Register the Driver

2)load the Driver

3)getthe connection

4)create the statement

5)Execute the query

6)fetch the results with ResultSet

Whatare the steps involved in establishing a connection?

Thisinvolves two steps: (1) loading the driver and (2) making theconnection.

Howcan you load the drivers?

Loadingthe driver or drivers you want to use is very simple and involvesjust one line of code. If, for example, you want to use the JDBC-ODBCBridge driver, the following code will load it:

Eg. Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

Yourdriver documentation will give you the class name to use. Forinstance, if the class name is jdbc.DriverXYZ , you would load thedriver with the following line of code:

E.g. Class.forName("jdbc.DriverXYZ");

WhatClass.forName will do while loading drivers?

Itis used to create an instance of a driver and registerit with the DriverManager.When you have loaded a driver, it is available for making aconnection with a DBMS.

Howcan you make the connection?

Inestablishing a connection is to have the appropriate driver connectto the DBMS. The following line of code illustrates the general idea:

E.g.

Stringurl = "jdbc:odbc:Fred";

Connectioncon = DriverManager.getConnection(url, "Fernanda", "J8");

Howcan you create JDBC statements?

AStatement object is what sends your SQL statement to the DBMS. Yousimply create a Statement object and then execute it, supplying theappropriate execute method with the SQL statement you want to send.For a SELECT statement, the method to use is executeQuery. Forstatements that create or modify tables, the method to use isexecuteUpdate. E.g. It takes an instance of an active connection tocreate a Statement object. In the following example, we use ourConnection object con to create the Statement object stmt :

Statementstmt = con.createStatement();

Howcan you retrieve data from the ResultSet?

FirstJDBC returns results in a ResultSet object, so we need to declare aninstance of the class ResultSet to hold our results. The followingcode demonstrates declaring the ResultSet object rs.

E.g.

ResultSetrs = stmt.executeQuery("SELECT COF\_NAME, PRICE FROM COFFEES");

Second:

Strings = rs.getString("COF\_NAME");

Themethod getString is invoked on the ResultSet object rs , so getStringwill retrieve (get) the value stored in the column COF\_NAME in thecurrent row of rs

Whatare the different types of Statements?

1.CreateStatement : For Simple statement used for static query.

2.PreparedStatement:For a runtime / dynamic query .Where String is a dynamic query youwant to execute

3.CallableStatement(Use prepareCall) : //For Stored procedure Callable statement, wheresql is stored procedure.

try

{

Connectionconn = DriverManager.getConnection("URL",'USER"."PWD");

Statementstmt = conn.createStatement();

PreparedStatementpstmt = conn.prepareStatement(String sql);

CallableStatementcstmt = conn.prepareCall(String sql);

}

catch(SQLException ee)

{

ee.printStackTrace();

}

Don'tforget all the above statements will throw the SQLException, so weneed to use try catch for the same to handle the exception.

Howcan you use PreparedStatement?

Thisspecial type of statement is derived from the more general class,Statement. If you want to execute a Statement object many times, itwill normally reduce execution time to use a PreparedStatement objectinstead. The advantage to this is that in most cases, this SQLstatement will be sent to the DBMS right away, where it will becompiled. As a result, the PreparedStatement object contains not justan SQL statement, but an SQL statement that has been precompiled.This means that when the PreparedStatement is executed, the DBMS canjust run the PreparedStatement 's SQL statement without having tocompile it first.

E.g. PreparedStatement updateSales = con.prepareStatement("UPDATECOFFEES SET SALES = ? WHERE COF\_NAME LIKE ?");

Howto call a Stored Procedure from JDBC?

Thefirst step is to create a CallableStatement object. As with Statementan and PreparedStatement objects, this is done with an openConnection object. A CallableStatement object contains a call to astored procedure;

E.g.

CallableStatementcs = con.prepareCall("{call SHOW\_SUPPLIERS}");

ResultSetrs = cs.executeQuery();

Howto Retrieve Warnings?

SQLWarningobjects are a subclass of SQLException that deal with database accesswarnings. Warnings do not stop the execution of an application, asexceptions do; they simply alert the user that something did nothappen as planned. A warning can be reported on a Connection object,a Statement object (including PreparedStatement and CallableStatementobjects), or a ResultSet object. Each of these classes has agetWarnings method, which you must invoke in order to see the firstwarning reported on the calling object

E.g.

SQLWarningwarning = stmt.getWarnings();

if(warning != null) {

while(warning != null) {

System.out.println("Message:" + warning.getMessage());

System.out.println("SQLState:" + warning.getSQLState());

System.out.print("Vendorerror code: ");

System.out.println(warning.getErrorCode());

warning= warning.getNextWarning();

}

}

Howto Make Updates to Updatable Result Sets?

Anothernew feature in the JDBC 2.0 API is the ability to update rows in aresult set using methods in the Java programming language rather thanhaving to send an SQL command. But before you can take advantage ofthis capability, you need to create a ResultSet object that isupdatable. In order to do this, you supply the ResultSet constantCONCUR\_UPDATABLE to the createStatement method.

E.g.

Connectioncon = DriverManager.getConnection("jdbc:mySubprotocol:mySubName");

Statementstmt = con.createStatement(ResultSet.TYPE\_SCROLL\_SENSITIVE,ResultSet.CONCUR\_UPDATABLE);

ResultSetuprs = ("SELECT COF\_NAME, PRICE FROM COFFEES");

\_\_\_\_\_\_\_€€€\_\_\_\_\_\_\_===€\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_€\_\_\_\_\_\_\_\_\_\_\_\_€€\_\_\_\_\_\_\_\_

SERVLETS

What is the servlet?

Servlets are modulesthat extend request/response-oriented servers, such as Java-enabledweb servers. For example, a servlet may be responsible for takingdata in an HTML order-entry form and applying the business logic usedto update a company's order database.

-Servlets are used toenhance and extend the functionality of Webserver.

-Servlets handles Java andHTML separately.

What are the uses ofServlets?

A servlet can handlemultiple requests concurrently, and can synchronize requests. Thisallows servlets to support systems such as on-line conferencing.Servlets can forward requests to other servers and servlets. Thusservlets can be used to balance load among several servers thatmirror the same content, and to partition a single logical serviceover several servers, according to task.

What are the charactersof Servlet?

AsServlet are written in java, they can make use of extensive power ofthe JAVA API,such as networking and URLaccess,multithreading,databaseconnectivity,RMI object serialization.

Efficient: The initilazation code for a servlet is executed only once, whenthe servlet is executed for the first time.

Robust : provide all the powerful features of JAVA, such as Exception handlingand garbage collection.

Portable:This enables easy portability across Web Servers.

Persistance: Increase the performance of the system by executing features dataaccess.

What is the differencebetween JSP and SERVLETS

Servlets: servlet tieup files to independently handle the static presentationlogic and dynamic business logic , due to this a changes made to anyfile requires recompilation of the servlet.

- The servlet isPre-Compile.

JSP: Facilities segregation of work profiles to Web-Developer andWeb-Designer , Automatically incorporates changes made to any file(PL & BL) , no need to recompile.

Web-Developer write thecode for Bussiness logic whereas Web-Designer designs the layout forthe WebPage by HTML & JSP.

- The JSP is Post-Compile.

What are the advantagesusing servlets than using CGI?

Servlets provide away to generate dynamic documents that is both easier to write andfaster to run. It is efficient, convenient, powerful, portable,secure and inexpensive. Servlets also address the problem of doingserver-side programming with platform-specific APIs. They aredeveloped with Java Servlet API, a standard Java extension.

What is the differencebetween servlets and applets?

Servlets are toservers. Applets are to browsers. Unlike applets, however, servletshave no graphical user interface.

What is the differencebetween GenericServlet and HttpServlet?

GenericServletis for servlets that might not use HTTP, like for instance FTPservice.As of only Http is implemented completely in HttpServlet. TheGenericServlet has a service() method that gets called when a clientrequest is made. This means that it gets called by both incomingrequests and the HTTP requests are given to the servlet as they are.

GenericServlet belongs tojavax.servlet package

GenericServlet is anabstract class which extends Object and implements Servlet,ServletConfig and java.io.Serializable interfaces.

The direct subclass toGenericServlet is HttpServlet.It is a protocol-independent servlet

What are the differencesbetween GET and POST service methods?

Get Method: UsesQuery String to send additional information to the server.

-QueryString is displayed on the client Browser.

Query String : Theadditional sequence of characters that are appended to the URL iacalled Query String. The length of the Query string is limited to255 characters.

-The amount of informationyou can send back using a GET is restricted as URLs can only be 1024characters.

POST Method: ThePost Method sends the Data as packets through a separate socketconnection. The complete transaction is invisible to the client. Thepost method is slower compared to the Get method because Data is sentto the server as separate packates.

--You can send much moreinformation to the server this way - and it's not restricted totextual data either. It is possible to send files and even binarydata such as serialized Java objects!

What is the servlet lifecycle?

In Servlet life cycles are,

init(),services(),destory().

Init( ): Is called by the Servlet container after the servlet has benInstantiated.

--Contains all informationcode for servlet and is invoked when the servlet is first loaded.

-The init( ) does notrequire any argument , returns a void and throws Servlet Exception.

-If init() executed at thetime of servlet class loading.And init() executed only for firstuser.

-You can Override thismethod to write initialization code that needs to run only once, suchas loading a driver , initializing values and soon, Inother case youcan leave normally blank.

Public voidinit(ServletConfig Config) throws ServletException

Service( ): is called by the Servlet container after the init method to allowthe servlet to respond to a request.

-Receives the request fromthe client and identifies the type of request and deligates them todoGet( ) or doPost( ) for processing.

Public voidservice(ServletRequest request,ServletResponce response) throwsServletException, IOException

Destroy( ): The Servlet Container calls the destroy( ) before removing aServlet Instance from Sevice.

-Executes only once when theServlet is removed from Server.

Public void destroy( )

If services() are bothfor get and post methods.

-So if u want to use postmethod in html page,we use doPost() or services() in servlet class.

-if want to use get methodsin html page,we can use doGet() or services() in servlet calss.

-Finally destory() is usedto free the object.

What is the differencebetween ServletContext and ServletConfig?

Both are interfaces.

Servlet Config():Theservlet engine implements the ServletConfig interface in order topass configuration information to a servlet. The server passes anobject that implements the ServletConfig interface to the servlet'sinit() method.

A ServletConfig objectpasses configuration information from the server to a servlet.ServletConfig also includes ServletContext object.

getParameter( ) ,getServletContext( ) , getServletConfig( ), GetServletName( )

Servlet Context(): TheServletContext interface provides information to servlets regardingthe environment in which they are running. It also provides standardway for servlets to write events to a log file.

ServletContext definesmethods that allow a servlet to interact with the host server. Thisincludes reading server-specific attributes, finding informationabout particular files located on the server, and writing to theserver log files. If there are several virtual servers running, eachone may return a different ServletContext.

getMIMEType( ) ,getResourse( ), getContext( ),getServerInfo( ),getServletContetName()

Can I invoke a JSP errorpage from a servlet?

Yes, you can invoke theJSP error page and pass the exception object to it from within aservlet. The trick is to create a request dispatcher for the JSPerror page, and pass the exception object as ajavax.servlet.jsp.jspException request attribute. However, note thatyou can do this from only within controller servlets.

If your servlet opens anOutputStream or PrintWriter, the JSP engine will throw the followingtranslation error:

java.lang.IllegalStateException:Cannot forward as OutputStream or Writer has already been obtained

Can I just abortprocessing a JSP?

Yes.Because your JSP is justa servlet method,you can just put (whereever necessary) a < %return; %>

What is a betterapproach for enabling thread-safe servlets and JSPs?SingleThreadModel Interface or Synchronization?

Although theSingleThreadModel technique is easy to use, and works well for lowvolume sites, it does not scale well. If you anticipate your users toincrease in the future, you may be better off implementing explicitsynchronization for your shared data. The key however, is toeffectively minimize the amount of code that is synchronzied so thatyou take maximum advantage of multithreading.

Also, note thatSingleThreadModel is pretty resource intensive from the server'sperspective. The most serious issue however is when the number ofconcurrent requests exhaust the servlet instance pool. In that case,all the unserviced requests are queued until something becomes free -which results in poor performance. Since the usage isnon-deterministic, it may not help much even if you did add morememory and increased the size of the instance pool.

If you want a servlet totake the same action for both GET and POST request, what should youdo?

Simply have doGet calldoPost, or vice versa.

Which code line must beset before any of the lines that use the PrintWriter?

setContentType() method mustbe set before transmitting the actual document.

How HTTP Servlet handlesclient requests?

An HTTP Servlethandles client requests through its service method. The servicemethod supports standard HTTP client requests by dispatching eachrequest to a method designed to handle that request.

What is the ServletInterface?

The centralabstraction in the Servlet API is the Servlet interface. All servletsimplement this interface, either directly or, more commonly, byextending a class that implements it such as HttpServlet.

Servlets-->GenericServlet-->HttpServlet-->MyServlet.

The Servlet interfacedeclares, but does not implement, methods that manage the servlet andits communications with clients. Servlet writers provide some or allof these methods when developing a servlet.

a servlet accepts acall from a client, it receives two objects. What are they?

ServeltRequest: whichencapsulates the communication from the client to the server.

ServletResponse: whichencapsulates the communication from the servlet back to the

Client.

ServletRequest andServletResponse are interfaces defined by the javax.servlet package.

What information thatthe ServletRequest interface allows the servlet access to?

Information such asthe names of the parameters passed in by the client, the protocol(scheme) being used by the client, and the names of the remote hostthat made the request and the server that received it. The inputstream, ServletInputStream.Servlets use the input stream to get data

from clients that useapplication protocols such as the HTTP POST and PUT methods.

What information thatthe ServletResponse interface gives the servlet methods for replyingto the client?

It Allows the servletto set the content length and MIME type of the reply. Provides anoutput stream, ServletOutputStream and a Writer through which theservlet can send the reply data.

Difference betweensingle thread and multi thread model servlet

A servlet thatimplements SingleThreadModel means that for every request, a singleservlet instance is created. This is not a very scalable solution asmost web servers handle multitudes of requests. A multi-threadedservlet means that one servlet is capable of handling many requestswhich is the way most servlets should be implemented.

A single thread model forservlets is generally used to protect sensitive data ( bank accountoperations ).

Single thread model meansinstance of the servlet gets created for each request recieved. Itsnot thread safe whereas in multi threaded only single instance ofthe servlet exists for what ever # of requests recieved. Its threadsafe and is taken care by the servlet container.

A servlet that implementsSingleThreadModel means that for every request, a single servletinstance is created. This is not a very scalable solution as mostweb servers handle multitudes of requests. A multi-threaded servletmeans that one servlet is capable of handling many requests whichis the way most servlets should be implemented.

A single thread model forservlets is generally used to protect sensitive data ( bank accountoperations ).

What is servlet contextand what it takes actually as parameters?

Servlet context isan object which is created as soon as the Servlet getsinitialized.Servlet context object is contained in Servlet Config.With the context object u can get access to specific

resource (like file) in theserver and pass it as a URL to be displayed as a next screen with thehelp of RequestDispatcher

eg :-

ServletContext app =getServletContext();

RequestDispatcher disp;

if(b==true)

disp =app.getRequestDispatcher

("jsp/login/updatepassword.jsp");

else

disp =app.getRequestDispatcher

("jsp/login/error.jsp");

this code will take user tothe screen depending upon the value of b.

in ServletContext u can alsoget or set some variables which u would

like to retreive in nextscreen.

eg

context.setAttribute("supportAddress","temp@temp.com");

Better yet, you could usethe web.xml context-param element to

designate the address, thenread it with the getInitParameter method

of ServletContext.

Can we call destroy()method on servlets from service method?

destroy() is a servletlife-cycle method called by servlet container to kill the instance ofthe servlet. "Yes". You can call destroy() from withinthe service(). It will do whatever logic you have in destroy()(cleanup, remove attributes, etc.) but it won't "unload"the servlet instance itself. That can only be done by the container

What is the use ofServletConfig and ServletContext..?

An interface thatdescribes the configuration parameters for a servlet. This is passedto the servlet when the web server calls its init() method. Note thatthe servlet should save the reference to the ServletConfig object,and define a getServletConfig() method to return it when asked. Thisinterface defines how to get the initialization parameters for theservlet and the context under which the servlet is running.

An interface thatdescribes how a servlet can get information about the server in whichit is running. It can be retrieved via the getServletContext() methodof the ServletConfig object.

What is differencebetween forward() and sendRedirect().. ? Which one is faster thenother and which works on server?

Forward( ) : javax.Servlet.RequestDispatcher interface.

-RequestDispatcher.forward() works on the Server.

-Theforward( ) works inside the WebContainer.

-The forward( ) restrictsyou to redirect only to a resource in the same web-Application.

-After executing theforward( ), the control will return back to the same method fromwhere the forward method was called.

-the forward( ) willredirect in the application server itself, it doesâ€™n come back tothe client.

- The forward( ) is fasterthan Sendredirect( ) .

To use the forward( ) of therequestDispatcher interface, the first thing to do is to obtainRequestDispatcher Object. The Servlet technology provides in threeways.

1. By using thegetRequestDispatcher( ) of the javax.Servlet.ServletContext interface, passing a String containing the path of the other resources, pathis relative to the root of the ServletContext.

RequestDispatcherrd=request.getRequestDispatcher(â€œsecondServletâ€);

Rd.forward(request,response);

2. getRequestDispatcher( )of the javax.Servlet.Requestinterface , the path is relative to current HtpRequest.

RequestDispatcherrd=getServletContext().getRequestDispatcher(â€œservlet/secondServletâ€);

Rd.forward(request,response);

3. By using thegetNameDispatcher( ) of the javax.Servlet.ServletContextinterface.

RequestDispatcherrd=getServletContext( ).getNameDispatcher(â€œsecondServletâ€);

Rd.forward(request,response);

Sendredirect( ) :javax.Servlet.Http.HttpServletResponce interface

-RequestDispatcher.SendRedirect() works on the browser.

-The SendRedirect( ) allowsyou to redirect trip to the Client.

-The SendRedirect( ) allowsyou to redirect to any URL.

-After executing theSendRedirect( ) the control will not return back to same method.

-The Client receives theHttp response code 302 indicating that temporarly the client is beingredirected to the specified location , if the specified location isrelative , this method converts it into an absolute URL beforeredirecting.

-The SendRedirect( ) willcome to the Client and go back,.. ie URL appending will happen.

Response. SendRedirect(â€œabsolute pathâ€);

Absolutepath â€“ other thanapplication , relative path - same application.

When you invoke aforward request, the request is sent to another resource on theserver, without the client being informed that a different resourceis going to process the request. This process occurs completely within the web container. When a sendRedirtect method is invoked, itcauses the web container to return to the browser indicating that anew URL should be requested. Because the browser issues a completelynew request any object that are stored as request attributes beforethe redirect occurs will be lost. This extra round trip a redirect isslower than forward.

do we have a constructorin servlet ? can we explictly provide a constructor in servletprogramme as in java program ?

We can have a constructor inservlet .

Session :A session is a group of activities that are performed by a user whileaccesing a particular website.

Session Tracking:The process of keeping track of settings across session is calledsession tracking.

Hidden Form Fields: Used to keep track of users by placing hidden fields in the form.

-The values that have beenentered in these fields are sent to the server when the user submitsthe Form.

URL-rewriting: this is a technique by which the URL is modified to include thesession ID of a particular user and is sent back to the Client.

-The session Id is used bythe client for subsequent transactions with the server.

Cookies: Cookies are small text files that are used by a webserver to keeptrack the Users.

Acookie is created by the server and send back to the client , thevalue is in the form of Key-value pairs. Aclient can accept 20cookies per host and the size of each cookie can be maximum of 4bytes each.

HttpSession: Every user who logs on to the website is autometacally associatedwith an HttpSession Object.

-The Servlet can use thisObject to store information about the users Session.

-HttpSession Object enablesthe user to maintain two types of Data.

ie State andApplication.

How to communicatebetween two servlets?

Two ways:

Forward or redirect fromone Servlet to another.

Load the Servlet fromServletContext and access methods.

How to get one Servlet'sContext Information in another Servlet?

Access or load the Servletfrom the Servlet Context and access the Context Information

The following codesnippet demonstrates the invocation of a JSP error page from withina controller servlet:

protected voidsendErrorRedirect(HttpServletRequest request,

HttpServletResponseresponse, String errorPageURL, Throwable e) throws

ServletException,IOException {

request.setAttribute("javax.servlet.jsp.jspException", e);

getServletConfig().getServletContext().

getRequestDispatcher(errorPageURL).forward(request,response);

}

public voiddoPost(HttpServletRequest request, HttpServletResponse response)

{

try {

// do something

} catch (Exception ex) {

try {

sendErrorRedirect(request,response,"/jsp/MyErrorPage.jsp",ex);

} catch (Exception e) {

e.printStackTrace();

}

} }

\_\_\_\_€€€\_\_\_€€€€\_\_\_€€€\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

JSP (JavaServerPages)

Why JSP Technology?

Servlets are good at runninglogic.

Not so good at producinglarge amounts of output

out.write() is ugly

JSP pages are great atproducing lots of textual output

Not so good at lots of logic

<% %> isugly

How does it Work

JSP page

Mixture of text, Script anddirectives

Text could be text/ html,text/ xml or text/ plain

JSP engine

Compiles page to servlet

Executes servletâ€™sservice() method

Sends text back to caller

Page is

Compiled once

Executed many times

Anatomy of a JSP

<%@ page language=javacontentType=text/html %>

<html>

<body bgcolor=white>

<jsp:useBean id=greetingclass=com.pramati.jsp.beans.GreetingBean>

<jsp:setPropertyname=greeting property=\*/>

</jsp:userBean>

The following informationwas saved:

User Name:

<jsp:getPropertyname=greeting property=userName/>

Welcome!

</body>

</html>

JSPElements

Directive Elements :Information about the page

Remains same betweenrequests

E.g., scripting languageused

Action Elements : Takeaction based on info required at request-time

Standard

Custom (Tags and TagLibraries)

Scripting Elements

Add pieces of code togenerate output based on conditions

Directives

Global information used bythe JSP engine

Of form <%@directive attr\_ list %>

Or <jsp:directive. directive attr\_ list />

Directive could be

Page

Include

Taglib

E. g.,

<%@ page info= written byDevelopMentor %>

<jsp: directive. pageimport= java. sql.\* />

<%@ include file =\somefile. txt %>

<%@ taglib uri = tagsprefix= foo %>

Actions Within a JSPPage

Specifies an action to becarried out by the JSP engine

Standard or custom

Standard must be implementedby all engines

Custom defined in taglibraries

Standard actions â€˜scopedâ€™by â€˜jspâ€™ namespace

Have name and attributes

<jsp: useBean id= clockclass= java.util.Date />

<ul> The current dateat the server is:

<li> Date: <jsp:getProperty name=clock property=date />

<li> Month: <jsp:getProperty name=clock property=month />

</ul>

Standard JSP Actions :

jsp:useBean

jsp:getProperty

jsp:setProperty

jsp:include

jsp:forward

jsp:param

jsp:plugin

Scriptlets

Of form <%/\* code goes here\*/ %>

Gets copied into \_jspServicemethod ofgenerated servlet

Any valid Java code can gohere

CODE: OUTPUT

<% int j; %> <value> 0</ value>

<% for (j = 0; j < 3;j++) {%> <value> 1</ value>

<value> <value> 2</ value>

<% out. write(+ j); %>

</ value><% } %>

Declarations (<%! â€¦%>)

Used to declare class scopevariables or methods

<%!int j = 0; %>

Gets declared at class-level scope in the generated servlet

public class SomeJSP extendsHttpServlet implements HttpJspPage {

â€¦

int j = 0;

void \_jspService(â€¦) {}

}

Declarations (<%! â€¦%>)

Used to declare class scopevariables or methods

<%! int j = 0; %>

Gets declared at class-level scope in the generated servlet

public class SomeJSP extendsHttpServlet implements HttpJspPage {

â€¦

int j = 0;

void \_jspService(â€¦) {}

}

JSPto Servlet Translation

<%@ pageimport=javax.ejb.\*,javax.naming.\*,java.rmi.\* ,java.util.\* %>

<HTML><HEAD><TITLE>Hello.jsp</TITLE></HEAD><BODY>

<%String checking =null;

String name = null;

checking =request.getParameter(catch);

if (checking != null) {

name =request.getParameter(name);%>

<b> Hello <%=name%>

<%} %>

<FORM METHOD='POST'action=Hello.jsp>

<table width=500cellspacing=0 cellpadding=3 border=0>

<caption>Enter yourname</caption>

<tr><td><b>Name</b></td><td><INPUTsize=20 maxlength=20 TYPE=text NAME=name></td></tr>

</table>

<INPUT TYPE='SUBMIT'NAME='Submit' VALUE='Submit'>

<INPUT TYPE='hidden'NAME='catch' VALUE='yes'>

</FORM></BODY></HTML>

GeneratedServletâ€¦

public void\_jspService(HttpServletRequest request ,

HttpServletResponseresponse)

throwsServletException ,IOException {

out.write(<HTML><HEAD><TITLE>Hello.jsp</TITLE></HEAD><BODY>);

String checking = null;

String name = null;

checking =request.getParameter(catch);

if (checking != null) {

name =request.getParameter(name);

out.write(\r\n\t\t<b>Hello );

out.print(name);

out.write(\r\n\t\t);

}

out.write(\r\n\t\t<FORMMETHOD='POST' action=

+\Hello.jsp\>\r\n\t\t\t<tablewidth=\500\ cellâ€¦â€¦â€¦â€¦â€¦â€¦â€¦â€¦â€¦â€¦â€¦..

}

}

Tags& Tag Libraries

What Is a Tag Library?

JSP technology has a set ofpre- defined tags

<jsp: useBean â€¦/>

These are HTML like butâ€¦

â€¦ have limitedfunctionality

Can define new tags

Look like HTML

Can be used by page authors

Java code is executed whentag is encountered

Allow us to keep Java codeoff the page

Better separation of contentand logic

May Have Tags Toâ€¦

Process an SQL command

Parse XML and output HTML

Automatically call into anEJB component (EJB â„¢ technology- based component)

Get called on every requestto initialize script variables

Iterate over a ResultSet anddisplay the output in an HTML table

PrimaryTag Classes (javax.servlet.jsp.tagext.Tag)

Simple Tag Example :

<%@ tagliburi=/WEB-INF/mylib.tld prefix=test %>

<html><bodybgcolor=white>

<test:hello name=Robert/>

</body> </html>

public class HelloTagextends TagSupport {

private String name =World;

public voidsetName(String name) { this.name = name; }

public int doEndTag() {pageContext.getOut().println(Hello + name); }

}

mylib.tld

<taglib>â€¦â€¦

<tag><name>hello</name>

<tagclass>com.pramati.HelloTag</tagclass>

<bodycontent>empty</bodycontent>

<attribute><name>name</name></attribute>

</tag>

</taglib>

How Tag Handler methods areinvoked :

<prefix:tagName

attr1=value1------------setAttr1(value1)

attr2=value2------------setAttr2(value2)

>------------doStartTag()

This tags's body

</prefix:tagName>------------doEndTag()

Implementation of JSP pagewill use the tag handler for each â€˜actionâ€™ on page.

Summary

The JSP specification is a powerful system for creating structuredweb content

JSP technology allows non- programmers to develop dynamic web pages

JSP technology allows collaboration between programmers and pagedesigners when building web applications

JSP technology uses the Java programming language as the scriptlanguage

The generated servlet can be managed by directives

JSP components can be usedas the view in the MVC architecture

Authors using JSP technologyare not necessarily programmers using Java technology

Want to keep Java code off aJSP Page

Custom actions (taglibraries) allow the use of elements as a replacement for Java code

What is JSP- JavaServerPages ?

JavaServer Pages. Aserver-side technology, JavaServer pages are an extension to the Javaservlet technology that was developed by Sun. JSPs have dynamicscripting capability that works in tandem with HTML code, separatingthe page logic from the static elements -- the actual design anddisplay of the page. Embedded in the HTML page, the Java source codeand its extensions help make the HTML more functional, being used indynamic database queries, for example. JSPs are not restricted to anyspecific platform or server.

Jsp contains both static anddynamic resources at run time.Jsp extends web server functionalities

What are advantages ofJSP

whenever there is a changein the code, we dont have to recompile the jsp. it automatically doesthe compilation. by using custom tags and tag libraries the length ofthe java code is reduced.

What is the differencebetween include directive & jsp:include action

include directive ():if the file includes static text if the file is rarely changed (theJSP engine may not recompile the JSP if this type of included file ismodified) .

if you have a common codesnippet that you can reuse across multiple pages (e.g. headers andfooters)

jsp:include : forcontent that changes at runtime .to select which content to render atruntime (because the page and src attributes can take runtimeexpressions) for files that change often JSP:includenull

What are Custom tags. Whydo you need Custom tags. How do you create Custom tag

1) Custom tags are thosewhich are user defined.

2) Inorder to separate thepresentation logic in a separate class rather than keeping in jsppage we can use custom tags.

3) Step 1 : Build a classthat implements the javax.servlet.jsp.tagext.Tag interface asfollows. Compile it and place it under the web-inf/classes directory(in the appropriate package structure).

package examples;

import java.io.\*; //// THISPROGRAM IS EVERY TIME I MEAN WHEN U REFRESH THAT PARTICULAR CURRENTDATE THIS CUSTOM TAG WILL DISPLAY

import javax.servlet.jsp.\*;

importjavax.servlet.jsp.tagext.\*;

public class ShowDateTagimplements Tag {

private PageContextpageContext;

private Tag parent;

public int doStartTag()throws JspException {

return SKIP\_BODY; }

public int doEndTag()throws JspException {

try {

pageContext.getOut().write(+ new java.util.Date());

} catch (IOException ioe) {

throw newJspException(ioe.getMessage());

}

return EVAL\_PAGE; }

public void release() {

}

public voidsetPageContext(PageContext page) {

this.pageContext = page;

}

public void setParent(Tagtag) {

this.parent = tag;

}

public Tag getParent() {

return this.parent; } }

Step 2:Now we need todescribe the tag, so create a file called taglib.tld and place itunder the web-infdirectory.http://java.sun.com/j2ee/dtds/web-jsptaglibrary\_1\_1.dtd>1.0 1.1

myTaghttp://www.mycompany.com/taglib My own tag library showDateexamples.ShowDateTag Show the current date

Step 3 : Now we need to tellthe web application where to find the custom tags, and how they willbe referenced from JSP pages. Edit the web.xml file under theweb-inf directory and insert the following XMLfragement.http://www.mycompany.com/taglib /WEB-INF/taglib.tld

Step 4 : And finally,create a JSP page that uses the custom tag.Now restart the server andcall up the JSP page! You should notice that every time the page isrequested, the current date is displayed in the browser. Whilstthis doesn't explain what all the various parts of the tag are for(e.g. the tag description, page context, etc) it should get yougoing. If you use the tutorial (above) and this example, you shouldbe able to grasp what's going on! There are some methods in contextobject with the help of which u can get the server (or servletcontainer) information.

Apart from all thiswith the help of ServletContext u can implementServletContextListener and then use the get-InitParametermethod toread context initialization parameters as the basis of data that willbe made available to all servlets and JSP pages.

What are the implicitobjects in JSP & differences between them

There are nine implicitobjects in JSP.

1. request: The request object represents httprequest that are trigged byservice( ) invocation. javax.servlet

2. response:Theresponse object represents the servers response to request.

javax.servlet

3. pageContext: The page context specifies the single entry point to many of thepage attributes and is the convient place to put shared data.

javax.servlet.jsp.pagecontext

4. session: the session object represents the session created by the currentuser.

javax.Servlet.http.HttpSession

5. application: the application object represents servlet context , obtained fromservlet configaration . javax.Servlet.ServletContext

6. out: the out object represents to write the out put stream .

javax.Servlet.jsp.jspWriter

7. Config:the config object represents the servlet config interface from thispage,and has scope attribute. javax.Servlet.ServletConfig

8. page: The object is th eInstance of page implementation servlet classthat are processing the current request.

java.lang.Object

9. exception: These are used for different purposes and actually u no need tocreate these objects in JSP. JSP container will create these objectsautomatically.

java.lang.Throwable

You can directly use theseobjects.

Example:

If i want to put my usernamein the session in JSP.

JSP Page: In the aboutpage, i am using session object. But this session object is notdeclared in JSP file, because, this is implicit object and it will becreated by the jsp container.

If u see the java file forthis jsp page in the work folder of apache tomcat, u will find theseobjects are created.

What is jsp:usebean. Whatare the scope attributes & difference between these attributes

page, request, session,application

What is differencebetween scriptlet and expression

With expressionsin JSP, the results of evaluating the expression are converted to astring and directly included within the output page. Typicallyexpressions are used to display simple values of variables or returnvalues by invoking a bean's getter methods. JSP expressions beginwithin tags and do not include semicolons:

Butscriptletcan contain any number of language statements, variable or methoddeclarations, or expressions that are valid in the page scriptinglanguage. Within scriptlet tags, you can declare variables or methodsto use later in the file, write expressions valid in the pagescripting language,use any of the JSP implicit objects or any objectdeclared with a scriplet.

What is Declaration

Declarationis used in JSP to declare methods and variables.To add a declaration,you must use the sequences to enclose your declarations.

How do you connect to thedatabase from JSP

To be precise toconnect jdbc from jsp is not good idea ofcourse if ur working ondummy projects connecting to msaccess u can very well use the sameconnection objects and methods in ur scriplets and define urconnection object in init() method.

But if its realtime u can use DAO design patterns which is widely used. for exampleu write all ur connection object and and sql quires in a defienedmethod later use transfer object [TO ]which is all ur fields haveget/set methods and call it in business object[BO] so DAO is accessdwith precaution as it is the crucial. Finally u define java beanwhich is a class holding get/set method implementing serializationthus the bean is called in the jsp. So never connect to jdbc directlyfrom client side since it can be hacked by any one to get ur passwordor credit card info.

How do you call storedprocedures from JSP

By using callable statementwe can call stored procedures and functions from the database.

How do you restrict pageerrors display in the JSP page

set isErrorPage=false

How do you pass controlfrom one JSP page to another

we can forward control toaother jsp using jsp action tags forward or incllude

How do I have theJSP-generated servlet subclass my own custom servlet class, insteadof the default?

One should be verycareful when having JSP pages extend custom servlet classes asopposed to the default one generated by the JSP engine. In doing so,you may lose out on any advanced optimization that may be provided bythe JSPengine. In any case, your new superclass has to fulfill thecontract with the JSPengine by: Implementing the HttpJspPageinterface, if the protocol used is HTTP, or implementing JspPageotherwise Ensuring that all the methods in the Servlet interface aredeclared final Additionally, your servlet superclass also needs to dothe following:

The service() method has toinvoke the \_jspService() method

The init() method has toinvoke the jspInit() method

The destroy() method has toinvoke jspDestroy()

If any of the aboveconditions are not satisfied, the JSP engine may throw a translationerror. Once the superclass has been developed, you can have your JSPextend it as follows:

<%@ pageextends=packageName.ServletName %>

How does a servletcommunicate with a JSP page?

The following codesnippet shows how a servlet instantiates a bean and initializes itwith FORM data posted by a browser. The bean is then placed into therequest, and the call is then forwarded to the JSP page, Bean1.jsp,by means of a request dispatcher for downstream processing.

public void doPost(HttpServletRequest request, HttpServletResponse response)

{

try {

govi.FormBean f = newgovi.FormBean();

String id =request.getParameter(id);

f.setName(request.getParameter(name));

f.setAddr(request.getParameter(addr));

f.setAge(request.getParameter(age));

//use the id to compute

//additional bean propertieslike info

//maybe perform a db query,etc.

// . . .

f.setPersonalizationInfo(info);

request.setAttribute(fBean,f);

getServletConfig().getServletContext().getRequestDispatcher(/jsp/Bean1.jsp).forward(request, response);

} catch (Exception ex) {

. . .

} }

The JSP page Bean1.jsp canthen process fBean, after first extracting it from the defaultrequest scope via the useBean action.

jsp:useBean id=fBeanclass=govi.FormBean scope=request/

jsp:getProperty name=fBeanproperty=name /

jsp:getProperty name=fBeanproperty=addr /

jsp:getProperty name=fBeanproperty=age /

jsp:getProperty name=fBeanproperty=personalizationInfo /

Is there a way I can setthe inactivity lease period on a per-session basis?

Typically, a defaultinactivity lease period for all sessions is set within your JSPengineadmin screen or associated properties file. However, if your JSPengine supports the Servlet 2.1 API, you can manage the inactivitylease period on a per-session basis. This is done by invoking theHttpSession.setMaxInactiveInterval() method, right after the sessionhas been created.

For example:

<%session.setMaxInactiveInterval(300); %>

would reset the inactivityperiod for this session to 5 minutes. The inactivity interval is setin seconds.

How can I set a cookieand delete a cookie from within a JSP page?

A cookie, mycookie, can bedeleted using the following scriptlet:

<%

//creating a cookie

Cookie mycookie = newCookie(aName,aValue);

response.addCookie(mycookie);

//delete a cookie

Cookie killMyCookie = newCookie(mycookie, null);

killMyCookie.setMaxAge(0);

killMyCookie.setPath(/);

response.addCookie(killMyCookie);

%>

How can I declare methodswithin my JSP page?

You can declare methodsfor use within your JSP page as declarations. The methods can then beinvoked within any other methods you declare, or within JSPscriptlets and expressions.

Do note that you do not havedirect access to any of the JSP implicit objects like request,response, session and so forth from within JSP methods. However, youshould be able to pass any of the implicit JSP variables asparameters to the methods you declare.

For example:

<%!

public StringwhereFrom(HttpServletRequest req) {

HttpSession ses =req.getSession();

...

return req.getRemoteHost();

}

%>

<%

out.print(Hi there, I seethat you are coming in from );

%>

<%= whereFrom(request) %>

Another Example:

file1.jsp:

<%@pagecontentType=text/html%>

<%!

public void test(JspWriterwriter) throws IOException{

writer.println(Hello!);

}

%>

file2.jsp

<%@includefile=file1.jsp%>

<html>

<body>

<%test(out);% >

</body>

</html>

How can I enable sessiontracking for JSP pages if the browser has disabled cookies?

We know thatsession tracking uses cookies by default to associate a sessionidentifier with a unique user. If the browser does not supportcookies, or if cookies are disabled, you can still enable sessiontracking using URL rewriting. URL rewriting essentially includes thesession ID within the link itself as a name/value pair. However, forthis to be effective, you need to append the session ID for each andevery link that is part of your servlet response. Adding the sessionID to a link is greatly simplified by means of of a couple ofmethods: response.encodeURL() associates a session ID with a givenURL, and if you are using redirection, response.encodeRedirectURL()can be used by giving the redirected URL as input. Both encodeURL()and encodeRedirectedURL() first determine whether cookies aresupported by the browser; if so, the input URL is returned unchangedsince the session ID will be persisted as a cookie.

Consider thefollowing example, in which two JSP files, say hello1.jsp andhello2.jsp, interact with each other. Basically, we create a newsession within hello1.jsp and place an object within this session.The user can then traverse to hello2.jsp by clicking on the linkpresent within the page.Within hello2.jsp, we simply extract theobject that was earlier placed in the session and display itscontents. Notice that we invoke the encodeURL() within hello1.jsp onthe link used to invoke hello2.jsp; if cookies are disabled, thesession ID is automatically appended to the URL, allowing hello2.jspto still retrieve the session object. Try this example first withcookies enabled. Then disable cookie support, restart the brower, andtry again. Each time you should see the maintenance of the sessionacross pages. Do note that to get this example to work with cookiesdisabled at the browser, your JSP engine has to support URLrewriting.

hello1.jsp

<%@ page session=true %>

<%

Integer num = newInteger(100);

session.putValue(num,num);

String url=response.encodeURL(hello2.jsp);

%>

<ahref='<%=url%>'>hello2.jsp</a>

hello2.jsp

<%@ page session=true %>

<%

Integer i= (Integer)session.getValue(num);

out.println(Num value insession is +i.intValue());

How do I use a scriptletto initialize a newly instantiated bean?

A jsp:useBeanaction mayoptionally have a body. If the body is specified, its contents willbe automatically invoked when the specified bean is instantiated.Typically, the body will contain scriptlets or jsp:setProperty tagsto initialize the newly instantiated bean, although you are notrestricted to using those alone. The following example shows thetoday property of the Foo bean initialized to the current date whenit is instantiated. Note that here, we make use of a JSP expressionwithin the jsp:setProperty action.

<jsp:useBean id=fooclass=com.Bar.Foo >

<jsp:setProperty name=fooproperty=today

value=<%=java.text.DateFormat.getDateInstance().format(newjava.util.Date())

%>/ >

<%-- scriptlets callingbean setter methods go here --%>

</jsp:useBean >

How does JSP handlerun-time exceptions?

You can use the errorPageattribute of the page directive to have uncaught runtime exceptionsautomatically forwarded to an error processing page.

For example:

<%@ pageerrorPage=error.jsp %>

redirects the browser to theJSP page error.jsp if an uncaught exception is encountered duringrequest processing. Within error.jsp, if you indicate that it is anerror-processing page, via the directive:

<%@ page isErrorPage=true%>

the Throwable objectdescribing the exception may be accessed within the error page viathe exception implicit object.

Note: You must always use arelative URL as the value for the errorPage attribute.

How do I prevent theoutput of my JSP or Servlet pages from being cached by the browser?

You will need to setthe appropriate HTTP header attributes to prevent the dynamic contentoutput by the JSP page from being cached by the browser. Just executethe following scriptlet at the beginning of your JSP pages to preventthem from being cached at the browser. You need both the statementsto take care of some of the older browser versions.

<%

response.setHeader(Cache-Control,no-store);//HTTP 1.1

response.setHeader(Pragma,no-cache);//HTTP 1.0

response.setDateHeader(Expires, 0); //prevents caching at the proxy server

%>

How do I use commentswithin a JSP page

You can use JSP-stylecomments to selectively block out code while debugging or simply tocomment your scriptlets. JSP comments are not visible at the client.

For example:

<%-- the scriptlet is nowcommented out

<%

out.println(Hello World);

%> --%>

You can also use HTML-stylecomments anywhere within your JSP page. These comments are visible atthe client. For example:

<!-- (c) 2004javagalaxy.com -->

Of course, you can also usecomments supported by your JSP scripting language within yourscriptlets. For example, assuming Java is the scripting language, youcan have:

<%

//some comment

/\*\*

yet another comment \*\*/ %>

Can I stop JSP executionwhile in the midst of processing a request?

Yes. Preemptivetermination of request processing on an error condition is a good wayto maximize the throughput of a high-volume JSP engine. The trick(asuming Java is your scripting language) is to use the returnstatement when you want to terminate further processing. For example,consider:

<% if(request.getParameter(foo) != null) {

// generate some html orupdate bean property

} else {

/\* output some error messageor provide redirection back to the input form after creating amemento bean updated with the 'valid' form elements that were input.This bean can now be used by the previous form to initialize theinput elements that were valid then, return from the body of the\_jspService() method to terminate further processing \*/

return;

}

%>

Is there a way toreference the this variable within a JSP page?

Yes, there is. UnderJSP 1.0, the page implicit object is equivalent to this, and returnsa reference to the servlet generated by the JSP page.

How do I perform browserredirection from a JSP page?

You can use the responseimplicit object to redirect the browser to a different resource, as:

response.sendRedirect(http://www.exforsys.com/path/error.html);

You can also physicallyalter the Location HTTP header attribute, as shown below:

<%

response.setStatus(HttpServletResponse.SC\_MOVED\_PERMANENTLY);

String newLocn =/newpath/index.html;

response.setHeader(Location,newLocn);

%>

You can also use the:<jsp:forward page=/newpage.jsp /> Also note that you can onlyuse this before any output has been sent to the client. I beleve thisis the case with the response.sendRedirect() method as well. If youwant to pass any paramateres then you can pass using

<jsp:forwardpage=/servlet/login> <jsp:param name=username value=HARI /></jsp:forward>

How do I include staticfiles within a JSP page?

Answer Static resourcesshould always be included using the JSP include directive. This way,the inclusion is performed just once during the translation phase.The following example shows the syntax:

<%@ includefile=copyright.html %>

Do note that you shouldalways supply a relative URL for the file attribute. Although you canalso include static resources using the action, this is not advisableas the inclusion is then performed for each and every request.

What JSP lifecyclemethods can I override?

You cannotoverridethe \_jspService() method within a JSP page. You can however, overridethe jspInit() and jspDestroy() methods within a JSP page. jspInit()can be useful for allocating resources like database connections,network connections, and so forth for the JSP page. It is goodprogramming practice to free any allocated resources withinjspDestroy().

The jspInit() andjspDestroy() methods are each executed just once during the lifecycleof a JSP page and are typically declared as JSP declarations:

<%! public voidjspInit() {

. . . }

%>

<%!

public void jspDestroy() {

. . . }

%>

Can a JSP page processHTML FORM data?

Yes. However, unlikeservlets, you are not required to implement HTTP-protocol specificmethods like doGet() or doPost() within your JSP page. You can obtainthe data for the FORM input elements via the request implicit objectwithin a scriptlet or expression as:

<%

String item =request.getParameter(item);

int howMany = newInteger(request.getParameter(units)).intValue();

%>

or

<%=request.getParameter(item) %>

How do I mix JSP and SSI#include?

If you're just includingraw HTML, use the #include directive as usual inside your .jsp file.

<!--#includefile=data.inc-->

Butit's a little trickier if you want the server to evaluate any JSPcode that's inside the included file. If your data.inc file containsjsp code you will have to use <%@ vinclude=data.inc %> The<!--#include file=data.inc--> is used for including non-JSPfiles.

How can I implement athread-safe JSP page?

You can make your JSPsthread-safe by having them implement the SingleThreadModelinterface. This is done by adding the directive

<%@ pageisThreadSafe=false % >within your JSP page.

How do I include staticfiles within a JSP page?

Static resources shouldalways be included using the JSP include directive. This way, theinclusion is performed just once during the translation phase. Thefollowing example shows the syntax: Do note that you should alwayssupply a relative URL for the file attribute. Although you can alsoinclude static resources using the action, this is not advisable asthe inclusion is then performed for each and every request.

How do you prevent theCreation of a Session in a JSP Page and why?

By default, a JSP pagewill automatically create a session for the request if one does notexist. However, sessions consume resources and if it is not necessaryto maintain a session, one should not be created. For example, amarketing campaign may suggest the reader visit a web page for moreinformation. If it is anticipated that a lot of traffic will hit thatpage, you may want to optimize the load on the machine by notcreating useless sessions.

What is the pagedirective is used to prevent a JSP page from automatically creating asession:

<%@ pagesession=false>

Is it possible to sharean HttpSession between a JSP and EJB? What happens when I change avalue in the HttpSession from inside an EJB?

You can pass theHttpSession as parameter to an EJB method, only if all objects insession are serializable.This has to be consider as passed-by-value,that means that it's read-only in the EJB. If anything is alteredfrom inside the EJB, it won't be reflected back to the HttpSession ofthe Servlet Container.The pass-byreference can be used between EJBsRemote Interfaces, as they are remote references. While it ISpossible to pass an HttpSession as a parameter to an EJB object, itis considered to be bad practice (1) in terms of object orienteddesign. This is because you are creating an unnecessary couplingbetween back-end objects (ejbs) and front-end objects (HttpSession).Create a higher-level of abstraction for your ejb's api. Rather thanpassing the whole, fat, HttpSession (which carries with it a bunch ofhttp semantics), create a class that acts as a value object (orstructure) that holds all the data you need to pass back and forthbetween front-end/back-end. Consider the case where your ejb needs tosupport a non-http-based client. This higher level of abstractionwill be flexible enough to support it. (1) Core J2EE design patterns(2001)

Can a JSP pageinstantiate a serialized bean?

No problem! The useBeanaction specifies the beanName attribute, which can be used forindicating a serialized bean. For example:

<jsp:useBean id=shoptype=shopping.CD beanName=CD />

<jsp:getPropertyname=shop property=album />

Acouple of important points to note. Although you would have to nameyour serialized file filename.ser, you only indicate filename as thevalue for the beanName attribute. Also, you will have to place yourserialized file within the WEB-INFjspbeans directory for it to belocated by the JSP engine.

Can you make use of aServletOutputStream object from within a JSP page?

No.You are supposed to make use of only a JSPWriter object (given to youin the form of the implicit object out) for replying to clients. AJSPWriter can be viewed as a buffered version of the stream objectreturned by response.getWriter(), although from an implementationalperspective, it is not. A page author can always disable the defaultbuffering for any page using a page directive as:

<%@ page buffer=none%>

Can we implementsinterface or extends class in JSP?

No ,we can't implements interface or extends class in JSP

What are the stepsrequired in adding a JSP Tag Libraries?

1. Create a TLD file andconfigure the required class Information.

2. Create the JavaImplementation Source extending the JSP Tag Lib Class (TagSupport).

3. Compile and package it asloosed class file or as a jar under lib folder in Web Archive Filefor Class loading.

4. Place the TLD file underthe WEB-INF folder.

5. Add reference to the taglibrary in the web.xml file.

# =========== STRUTS ============

Struts1.1

Introductionto MVC

Overviewof MVC Architecture

ApplyingMVC in Servlets and JSP

Viewon JSP

JSPModel 1 Architecture

JSPModel 2 Architecture

Limitationin traditional MVC approach

MVCModel 2 Architecture

Thebenefits

Applicationflow

Overviewof Struts Framework

Introductionto Struts Framework

StrutsArchitecture

FrontController Design Pattern

Controllerservlet - ActionServlet

Actionobjects

ActionForm objects

Actionmappings

Configuringweb.xml file and struts-config.xml file

Strutsâ€“ View components

CompositeView

Buildingpage from templates

Jsp:includeVs struts template mechanism

Beantags

Htmltags

Logictags

Templatetags

Strutsâ€“ Controller components

Actionclasses

ActionServlet

Strutsdata source

AdvancedStruts

AccessingApplication Resource File

Useof Tokens

AccessingIndexed properties

ForwardVs Redirect

Dynamiccreating Action Forwards

Struts1.1

DynaActionForm

DynaValidatorActionForm

ValidatingInput Data

Declarativeapproach

Using Struts Validator

Configuring theValidator

Specifying validation rules

Client side validation

Plugins

I18N(InternationalizatioN)

Specifyinga resource bundle

Generating Locale specific messages

Tiles

Introductionto MVC(Model View Controler)

Struts: Struts is an open source framework from Jakartha Project designedfor developing the web applications with Java SERVLET API and JavaServer Pages Technologies.Struts conforms the Model View Controllerdesign pattern. Struts package provides unified reusable components(such as action servlet) to build the user interface that can beapplied to any web connection. It encourages software developmentfollowing the MVC design pattern.

Overviewof MVC Architecture

TheMVC design pattern divides applications into three components:

TheModelmaintains the state and data that the application represents .

TheViewallows the display of information about the model to the user.

TheControllerallows the user to manipulate the application .

InStruts, the view is handled by JSPs and presentation components, themodel is represented by Java Beans and the controller uses Servletsto perform its action.

Bydeveloping a familiar Web-based shopping cart, you'll learn how toutilize the Model-View-Controller (MVC) design pattern and trulyseparate presentation from content when using Java Server Pages.

ApplyingMVC in Servlets and JSP

Manyweb applications are JSP-only or Servlets-only. With JSP, Java codeis embedded in the HTML code; with Servlets the Java code callsprintln methods to generate the HTML code. Both approaches have theiradvantages and drawbacks; Struts gathers their strengths to get thebest of their association.

Belowyou will find one example on registration form processing using MVCin Servlets and JSP:

Inthe above application Reg.jsp act as view accepts I/P from client andsubmits to Controller Servlet.

ControllerServlet validates the form data, if valid, stores the data into DB

Basedon the validation and DB operations Controller Servlet decides torespond either Confirm.jsp or Error.jsp to clientâ€™s browser.

Whenthe Error.jsp is responded, the page must include all the list oferrors with detailed description.

Theabove shown application architecture is the model for MVC.

IFMVC Model 2 wants to be implemented in your application businesslogic and model operations must be separated from controllerprogram.

Viewon JSP

Theearly JSP specification follows two approaches for buildingapplications using JSP technology. These two approaches are called asJSP Model 1 and JSP Model 2 architectures.

InModel 1architecture theJSP page is alone responsible for processing the incoming request andreplying back to the client. There is still separation ofpresentation from content, because all data access is performed usingbeans. Although the JSP Model 1 Architecture is more suitable forsimple applications, it may not be desirable for compleximplementations.

TheModel2 Architectureis an approach for serving dynamic content, since it combines the useof both Servlets and JSP. It takes advantages of the predominantstrengths of both technologies, using JSP to generate thepresentation layer and Servlets to perform process-intensive tasks.Here servlet acts as controllerandis in charge of request processing and the creation of any beans orobjects used by the JSP as well as deciding depending on the userâ€™sactions, which JSP page to forward the request to. Note that there isno processing logic within the JSP page itself; it is simplyresponsible for retrieving any objects or beans that may have beenpreviously created by the servlet, and extracting the dynamic contentfrom that servlet for insertion within static templates.

Limitationin traditional MVC approach

Themain limitation in the traditional MVC approach is, in that there isno separation of business logic (validation/ conditions/ anythingrelated to business rules) from controller (is responsible forcontrolling of the application flow by using static/dynamic requestdispatcher.

Client submits login request to servlet application

Servletapplication acts as controller it first decides to request validatoranother servlet program which is responsible for not null checking(business rule)

controlcomes to controller back and based on the validation response, ifthe response is positive, servlet controller sends the request tomodel

Modelrequests DB to verify whether the database is having the same username and password, If found login operation is successful

Beansare used to store if any data retrieved from the database and keptinto HTTPSession

Controllerthen gives response back to response JSP (view) which uses the beanobjects stored in HTTPSession object

andprepares presentation response on to the browser

Overviewof Struts Framework

Introductionto Struts Framework

Thegoal of this project is to provide an open source framework forbuilding Java web applications. The core of the Struts framework is aflexible control layer based on standard technologies like JavaServlets, JavaBeans, Resource Bundles, and XML, as well as variousJakarta Commons packages. Struts encourages application architecturesbased on the Model 2 approach, a variation of the classicModel-View-Controller(MVC) design paradigm.

Strutsprovides its own Controller component and integrates with othertechnologies to provide the Model and the View.

Forthe Model, Struts can interact with standard data accesstechnologies, like JDBC and EJB, as well as most any third-partypackages, like Hibernate, iBATIS, or Object Relational Bridge.

Forthe View, Struts works well with Java Server Pages, including JSTLand JSF, as well as Velocity Templates, XSLT, and other presentationsystems.

ForController,ActionServlet and ActionMapping - The Controllerportion of the application is focused on receiving requests from theclient deciding what business logic function is to be performed, andthen delegating responsibility for producing the next phase of theuser interface to an appropriate View component. In Struts, theprimary component of the Controller is a servlet of classActionServlet.This servlet is configured by defining a set of ActionMappings.An ActionMapping defines a paththat is matched against the request URI of the incoming request, andusually specifies the fully qualified class name of an Action class.Actions encapsulate the business logic, interpret the outcome, andultimately dispatch control to the appropriate View component tocreate the response.

TheStruts project was launched in May 2000 by Craig McClanahan toprovide a standard MVC framework to the Java community. In July 2001.

Inthe MVC design pattern, application flow is mediated by a centralController. The Controller delegatesâ€™ requests - in our case, HTTPrequests - to an appropriate handler. The handlers are tied to aModel, and each handler acts as an adapter between the request andthe Model. The Model represents, or encapsulates, an application'sbusiness logic or state. Control is usually then forwarded backthrough the Controller to the appropriate View. The forwarding can bedetermined by consulting a set of mappings, usually loaded from adatabase or configuration file. This provides a loose couplingbetween the View and Model, which can make applications significantlyeasier to create and maintain.

FrontController

Context

Thepresentation-tier request handling mechanism must control andcoordinate processing of each user across multiple requests. Suchcontrol mechanisms may be managed in either a centralized ordecentralized manner.

Problem

Thesystem requires a centralized access point for presentation-tierrequest handling to support the integration of system services,content retrieval, view management, and navigation. When the useraccesses the view directly without going through a centralizedmechanism,

Twoproblems may occur:

Eachview is required to provide its own system services, often resultingin duplicate code.

Viewnavigation is left to the views. This may result in commingled viewcontent and view navigation.

Additionally,distributed control is more difficult to maintain, since changes willoften need to be made in numerous places.

Solution:

Usea controller as the initial point of contact for handling a request.The controller manages the handling of the request, includinginvoking security services such as authentication and authorization,delegating business processing, managing the choice of an appropriateview, handling errors, and managing the selection of content creationstrategies.

Thecontroller provides a centralized entry point that controls andmanages Web request handling. By centralizing decision points andcontrols, the controller also helps reduce the amount of Java code,called scriptlets,embedded in the JavaServer Pages (JSP) page.

Centralizingcontrol in the controller and reducing business logic in the viewpromotes code reuse across requests. It is a preferable approach tothe alternative-embedding code in multiple views-because thatapproach may lead to a more error-prone, reuse-by-copy- and-pasteenvironment.

Typically,a controller coordinates with a dispatcher component. Dispatchers areresponsible for view management and navigation. Thus, a dispatcherchooses the next view for the user and vectors control to theresource. Dispatchers may be encapsulated within the controllerdirectly or can be extracted into a separate component.

Whilethe Front Controller pattern suggests centralizing the handling ofall requests, it does not limit the number of handlers in the system,as does a Singleton. An application may use multiple controllers in asystem, each mapping to a set of distinct services.

Controller : Thecontroller is the initial contact point for handling all requests inthe system. The controller may delegate to a helper to completeauthentication and authorization of a user or to initiate contactretrieval.

Dispatcher :

Adispatcher is responsible for view management and navigation,managing the choice of the next view to present to the user, andproviding the mechanism for vectoring control to this resource.

Adispatcher can be encapsulated within a controller or can be aseparate component working in coordination. The dispatcher provideseither a static dispatching to the view or a more sophisticateddynamic dispatching mechanism.

Thedispatcher uses the Request Dispatcher object (supported in theservlet specification) and encapsulates some additional processing.

Helper :

A helperis responsible for helping a view or controller complete itsprocessing. Thus, helpers have numerous responsibilities, includinggathering data required by the view and storing this intermediatemodel, in which case the helper is sometimes referred to as a valuebean. Additionally, helpers may adapt this data model for use by theview. Helpers can service requests for data from the view by simplyproviding access to the raw data or by formatting the data as Webcontent.

A viewmay work with any number of helpers, which are typically implementedas JavaBeans components (JSP 1.0+) and custom tags (JSP 1.1+).Additionally, a helper may represent a Command object, a delegate, oran XSL Transformer, which is used in combination with a stylesheet toadapt and convert the model into the appropriate form.

View :Aview represents and displays information to the client. The viewretrieves information from a model. Helpers support views byencapsulating and adapting the underlying data model for use in thedisplay.

ControllerServlet â€“ Action Servlet

Forthose of you familiar with MVC architecture, the ActionServletrepresents the C - the controller. The job of the controller is to:

processuser requests,

determinewhat the user is trying to achieve according to the request,

pulldata from the model (if necessary) to be given to the appropriateview, and

selectthe proper view to respond to the user.

TheStruts controller delegates most of this grunt work to the RequestProcessorand Action classes.

Inaddition to being the front controller for your application, theActionServlet instance also is responsible for initialization andclean-up of resources. When the controller initializes, it firstloads the application config corresponding to the "config"init-param. It then goes through an enumeration of all init-paramelements, looking for those elements who's name starts with config/.For each of these elements, Struts loads the configuration filespecified by the value of that init-param,and assigns a "prefix" value to that module's ModuleConfiginstance consisting of the piece of the init-paramname following "config/". For example, the module prefixspecified by the init-paramconfig/foowould be "foo". This is important to know, since this ishow the controller determines which module will be given control ofprocessing the request. To access the module foo, you would use a URLlike:

http://localhost:8080/myApp/foo/someAction.do

Foreach request made of the controller, the methodprocess(HttpServletRequest,HttpServletResponse)will be called. This method simply determines which module shouldservice the request and then invokes that module's RequestProcessor'sprocess method, passing the same request and response.

RequestProcessor :

TheRequestProcessor is where the majority of the core processing occursfor each request. Let's take a look at the helper functions theprocess method invokes in-turn:

processPath

Determinethe path that invoked us. This will be used later to retrieve anActionMapping.

processLocale

Selecta locale for this request, if one hasn't already been selected,and place it in the request.

processContent

Setthe default content type (with optional character encoding) forall responses if requested.

processNoCache

Ifappropriate, set the following response headers: "Pragma","Cache-Control", and "Expires".

processPreprocess

Thisis one of the "hooks" the RequestProcessor makesavailable for subclasses to override. The default implementationsimply returns true.If you subclass RequestProcessor and override processPreprocessyou should either return true(indicating process should continue processing the request) orfalse(indicating you have handled the request and the process shouldreturn)

processMapping

Determinethe ActionMapping associated with this path.

processRoles

Ifthe mapping has a role associated with it, ensure the requestinguser is has the specified role. If they do not, raise an error andstop processing of the request.

processActionForm

Instantiate(if necessary) the ActionForm associated with this mapping (ifany) and place it into the appropriate scope.

processPopulate

Populatethe ActionForm associated with this request, if any.

processValidate

Performvalidation (if requested) on the ActionForm associated with thisrequest (if any).

processForward

Ifthis mapping represents a forward, forward to the path specifiedby the mapping.

processInclude

Ifthis mapping represents an include, include the result of invokingthe path in this request.

processActionCreate

Instantiatean instance of the class specified by the current ActionMapping(if necessary).

processActionPerform

Thisis the point at which your action's performor executemethod will be called.

processForwardConfig

Finally,the process method of the RequestProcessor takes the ActionForwardreturned by your Action class, and uses to select the nextresource (if any). Most often the ActionForward leads to thepresentation page that renders the response.

Actionclass

TheActionclass defines two methods that could be executed depending on yourservlet environment:

publicActionForward execute(ActionMapping mapping,

ActionFormform,

ServletRequestrequest,

ServletResponseresponse)

throwsException;

publicActionForward execute(ActionMapping mapping,

ActionFormform,

HttpServletRequestrequest,

HttpServletResponseresponse)

throwsException;

Sincethe majority of Struts projects are focused on building webapplications, most projects will only use the "HttpServletRequest"version. A non-HTTP execute() method has been provided forapplications that are not specifically geared towards the HTTPprotocol.

Thegoal of an Actionclass is to process a request, via its executemethod, and return an ActionForwardobject that identifies where control should be forwarded (e.g. a JSP,Tile definition, Velocity template, or another Action) to provide theappropriate response. In the MVC/Model2 designpattern, a typical Actionclass will often implement logic like the following in its executemethod:

Validatethe current state of the user's session (for example, checking thatthe user has successfully logged on). If the Actionclass finds that no logon exists, the request can be forwarded tothe presentation page that displays the username and passwordprompts for logging on. This could occur because a user tried toenter an application "in the middle" (say, from abookmark), or because the session has timed out, and the servletcontainer created a new one.

Ifvalidation is not complete, validate the form bean properties asneeded. If a problem is found, store the appropriate error messagekeys as a request attribute, and forward control back to the inputform so that the errors can be corrected.

Performthe processing required to deal with this request (such as saving arow into a database). This canbe done by logic code embedded within the Actionclass itself, butshould generally be performed by calling an appropriate method of abusiness logic bean.

Updatethe server-side objects that will be used to create the next page ofthe user interface (typically request scope or session scope beans,depending on how long you need to keep these items available).

Returnan appropriate ActionForwardobject that identifies the presentation page to be used to generatethis response, based on the newly updated beans. Typically, you willacquire a reference to such an object by calling findForwardon either the ActionMappingobject you received (if you are using a logical name local to thismapping), or on the controller servlet itself (if you are using alogical name global to the application).

InStruts 1.0, Actions called a performmethod instead of the now-preferred executemethod. These methods use the same parameters and differ only inwhich exceptions they throw. The elder performmethod throws SerlvetExceptionand IOException.The new executemethod simply throws Exception.The change was to facilitate the Declarative Exception handlingfeature introduced in Struts 1.1.

Theperformmethod may still be used in Struts 1.1 but is deprecated. The Struts1.1 method simply calls the new executemethod and wraps any Exceptionthrown as a ServletException.

ActionForm class

AnActionForm represents an HTML form that the user interacts with overone or more pages. You will provide properties to hold the state ofthe form with getters and setters to access them. ActionForms can bestored in either the session (default) or request scopes. If they'rein the session it's important to implement the form's resetmethod to initialize the form before each use. Struts sets theActionForm's properties from the request parameters and sends thevalidated form to the appropriate Action's executemethod.

Whenyou code your ActionFormbeans, keep the following principles in mind:

TheActionFormclass itself requires no specific methods to be implemented. It isused to identify the role these particular beans play in the overallarchitecture. Typically, an ActionFormbean will have only property getter and property setter methods,with no business logic.

TheActionForm object also offers a standard validation mechanism. Ifyou override a "stub" method, and provide error messagesin the standard application resource, Struts will automaticallyvalidate the input from the form (using your method). See "AutomaticForm Validation"for details. Of course, you can also ignore the ActionFormvalidation and provide your own in the Action object.

Definea property (with associated getXxxand setXxxmethods) for each field that is present in the form. The field nameand property name must match according to the usual JavaBeansconventions (see the Javadoc for the java.beans.Introspectorclass for a start on information about this). For example, an inputfield named usernamewill cause the setUsernamemethod to be called.

Buttonsand other controls on your form can also be defined as properties.This can help determine which button or control was selected whenthe form was submitted. Remember, the ActionForm is meant torepresent your data-entry form, not just the data beans.

Thinkof your ActionForm beans as a firewall between HTTP and the Action.Use the validatemethod to ensure all required properties are present, and that theycontain reasonable values. An ActionForm that fails validation willnot even be presented to the Action for handling.

Youmay also place a bean instance on your form, and use nested propertyreferences. For example, you might have a "customer" beanon your ActionForm, and then refer to the property "customer.name"in your presentation page. This would correspond to the methodscustomer.getName()and customer.setName(stringName)on your customer bean. See the Tag Library Developer Guides for moreabout using nested syntax with the Struts JSP tags.

Caution:If you nest an existing bean instance on your form, think about theproperties it exposes. Any public property on an ActionForm thataccepts a single String value can be set with a query string. It maybe useful to place beans that can affect the business state inside athin "wrapper" that exposes only the properties required.This wrapper can also provide a filter to be sure runtime propertiesare not set to inappropriate values.

Actionclass Design guidelines

Rememberthe following design guidelines when coding Actionclasses:

Writecode for a multi-threaded environment - The controller servletcreates only one instance of your Actionclass, and uses this one instance to service all requests. Thus, youneed to write thread-safe Actionclasses. Follow the same guidelines you would use to writethread-safe Servlets. Here are two general guidelines that will helpyou write scalable, thread-safe Action classes:

OnlyUse Local Variables - The most important principle that aids inthread-safe coding is to use only local variables, not instancevariables, in your Actionclass. Local variables are created on a stack that is assigned (byyour JVM) to each request thread, so there is no need to worry aboutsharing them. An Actioncan be factored into several local methods, so long as all variablesneeded are passed as method parameters. This assures thread safety,as the JVM handles such variables internally using the call stackwhich is associated with a single Thread.

ConserveResources - As a general rule, allocating scarce resources andkeeping them across requests from the same user (in the user'ssession) can cause scalability problems. For example, if yourapplication uses JDBC and you allocate a separate JDBC connectionfor every user, you are probably going to run in some scalabilityissues when your site suddenly shows up on Slashdot. You shouldstrive to use pools and release resources (such as databaseconnections) prior to forwarding control to the appropriate Viewcomponent -- even if a bean method you have called throws anexception.

Don'tthrow it, catch it! - Ever used a commercial website only to have astack trace or exception thrown in your face after you've alreadytyped in your credit card number and clicked the purchase button?Let's just say it doesn't inspire confidence. Now is your chance todeal with these application errors - in the Actionclass. If your application specific code throws expections youshould catch these exceptions in your Action class, log them in yourapplication's log (servlet.log("Errormessage", exception))and return the appropriate ActionForward.

Itis wise to avoid creating lengthy and complex Action classes. If youstart to embed too much logic in the Actionclass itself, you will begin to find the Actionclass hard to understand, maintain, and impossible to reuse. Ratherthan creating overly complex Action classes, it is generally a goodpractice to move most of the persistence, and "business logic"to a separate application layer. When an Action class becomes lengthyand procedural, it may be a good time to refactor your applicationarchitecture and move some of this logic to another conceptual layer;otherwise, you may be left with an inflexible application which canonly be accessed in a web-application environment. Struts should beviewed as simply the foundation for implementing MVC in yourapplications. Struts provides you with a useful control layer, but itis not a fully featured platform for building MVC applications, soupto nuts.

TheMailReader example application included with Struts stretches thisdesign principle somewhat, because the business logic itself isembedded in the Actionclasses. This should be considered something of a bug in the designof the example, rather than an intrinsic feature of the Strutsarchitecture, or an approach to be emulated. In order to demonstrate,in simple terms, the different ways Struts can be used, theMailReader application does not always follow best practices.

Actionmapping implementation

Inorder to operate successfully, the Struts controller servlet needs toknow several things about how each request URI should be mapped to anappropriate Actionclass. The required knowledge has been encapsulated in a Java classnamed ActionMapping,the most important properties are as follows:

type- Fully qualified Java class name of the Action implementation classused by this mapping.

name- The name of the form bean defined in the config file that thisaction will use.

path- The request URI path that is matched to select this mapping. Seebelow for examples of how matching works and how to use wildcards tomatch multiple request URIs.

unknown- Set to trueif this action should be configured as the default for thisapplication, to handle all requests not handled by another action.Only one action can be defined as a default within a singleapplication.

validate- Set to trueif the validatemethod of the action associated with this mapping should be called.

forward- The request URI path to which control is passed when this mappingis invoked. This is an alternative to declaring a typeproperty.

WritingAction Mappings

Howdoes the controller servlet learn about the mappings you want? Itwould be possible (but tedious) to write a small Java class thatsimply instantiated new ActionMappinginstances, and called all of the appropriate setter methods. To makethis process easier, Struts uses the Jakarta Commons Digestercomponent to parse an XML-based description of the desired mappingsand create the appropriate objects initialized to the appropriatedefault values. See the JakartaCommons website formore information about the Digester.

Thedeveloper's responsibility is to create an XML file namedstruts-config.xmland place it in the WEB-INF directory of your application. Thisformat of this document is described by the Document Type Definition(DTD) maintained at

http://jakarta.apache.org/struts/dtds/struts-config\_1\_2.dtd.This chapter covers the configuration elements that you willtypically write as part of developing your application. There areseveral other elements that can be placed in the struts-config fileto customize your application. See "ConfiguringApplications" formore about the other elements in the Struts configuration file.

Thecontroller uses an internal copy of this document to parse theconfiguration; an Internet connection is not required for operation.

Theoutermost XML element must be <struts-config>.Inside of the <struts-config> element, there are threeimportant elements that are used to describe your actions:

<form-beans>

<global-forwards>

<action-mappings>

<form-beans>

Thissection contains your form bean definitions. Form beans aredescriptors that are used to create ActionForm instances at runtime.You use a <form-bean> element for each form bean, which has thefollowing important attributes:

name:A unique identifier for this bean, which will be used to referenceit in corresponding action mappings. Usually, this is also the nameof the request or session attribute under which this form bean willbe stored.

type:The fully-qualified Java classname of the ActionForm subclass to usewith this form bean.

<global-forwards>

This section contains your global forward definitions. Forwardsare instances of the ActionForward class returned from anActionForm's executemethod. These map logical names to specific resources (typicallyJSPs), allowing you to change the resource without changingreferences to it throughout your application. You use a <forward>element for each forward definition, which has the followingimportant attributes:

name:The logical name for this forward. This is used in your ActionForm'sexecutemethod to forward to the next appropriate resource. Example:homepage

path:The context relative path to the resource. Example: /index.jsp or/index.do

redirect:Trueor false(default). Should the ActionServlet redirect to the resource insteadof forward?

<action-mappings>

Thissection contains your action definitions. You use an <action>element for each of the mappings you would like to define. Mostaction elements will define at least the following attributes:

path:The application context-relative path to the action.

type:The fully qualified java classname of your Action class.

name:The name of your <form-bean>element to use with this action

Otheroften-used attributes include:

parameter:A general-purpose attribute often used by "standard"Actions to pass a required property.

roles:A comma-delimited list of the user security roles that can accessthis mapping.

Fora complete description of the elements that can be used with theactionelement, see the StrutsConfiguration DTD andthe ActionMappingdocumentation.

ActionMapping Example

Here'sa mapping entry based on the MailReader example application. TheMailReader application now uses DynaActionForms. But in this example,we'll show a conventinal ActionForm instead, to illustrate the usualworkflow. Note that the entries for all the other actions are leftout:

<struts-config>

<form-beans>

<form-bean

name="logonForm"

type="org.apache.struts.webapp.example.LogonForm"/>

</form-beans>

<global-forwards

type="org.apache.struts.action.ActionForward">

<forward

name="logon"

path="/logon.jsp"

redirect="false"/>

</global-forwards>

<action-mappings>

<action

path ="/logon"

type ="org.apache.struts.webapp.example.LogonAction"

name="logonForm"

scope="request"

input ="/logon.jsp"

unknown="false"

validate="true"/>

</action-mappings>

</struts-config>

Firstthe form bean is defined. A basic bean of class"org.apache.struts.webapp.example.LogonForm"is mapped to the logical name "logonForm".This name is used as a request attribute name for the form bean.

The"global-forwards"section is used to create logical name mappings for commonly usedpresentation pages. Each of these forwards is available through acall to your action mapping instance, i.e.mapping.findForward("logicalName").

Asyou can see, this mapping matches the path /logon(actually, because the MailReader example application uses extensionmapping, the request URI you specify in a JSP page would end in/logon.do).When a request that matches this path is received, an instance of theLogonActionclass will be created (the first time only) and used. The controllerservlet will look for a bean in request scope under key logonForm,creating and saving a bean of the specified class if needed.

Optionalbut very useful are the local "forward"elements. In the MailReader example application, many actions includea local "success" and/or "failure" forward aspart of an action mapping.

<!--Edit mail subscription -->

<action

path="/editSubscription"

type="org.apache.struts.webapp.example.EditSubscriptionAction"

name="subscriptionForm"

scope="request"

validate="false">

<forward

name="failure"

path="/mainMenu.jsp"/>

<forward

name="success"

path="/subscription.jsp"/>

</action>

Usingjust these two extra properties, the Action classes are almosttotally independent of the actual names of the presentation pages.The pages can be renamed (for example) during a redesign, withnegligible impact on the Action classes themselves. If the names ofthe "next" pages were hard coded into the Action classes,all of these classes would also need to be modified. Of course, youcan define whatever local forwardproperties makes sense for your own application.

TheStruts configuration file includes several other elements that youcan use to customize your application. See "ConfiguringApplications" fordetails.

UsingAction Mapping for pages

Frontingyour pages with ActionMappings is essentialwhen using modules, since doing so is the only way you involve thecontroller in the request -- and you want to! The controller puts theapplication configuration in the request, which makes available allof your module-specific configuration data (including which messageresources you are using, request-processor, datasources, and soforth).

Thesimplest way to do this is to use the forwardproperty of the ActionMapping:

<actionpath="/view" forward="/view.jsp"/>

Configuringstruts-config.xml file

TheBuildingController Componentschapter covered writing the form-bean and action-mapping portions ofthe Struts configuration file. These elements usually play animportant role in the development of a Struts application. The otherelements in Struts configuration file tend to be static: you set themonce and leave them alone.

These"static" configuration elements are:

controller

message-resources

plug-in

data-sources

Controllerconfiguration

The<controller>element allows you to configure the ActionServlet. Many of thecontroller parameters were previously defined by servletinitialization parameters in your web.xmlfile but have been moved to this section of struts-config.xmlin order to allow different modules in the same web application to beconfigured differently. For full details on available parameters seethe struts-config\_1\_2.dtdor the list below.

bufferSize- The size (in bytes) of the input buffer used when processing fileuploads. [4096] (optional)

className- Classname of configuration bean.[org.apache.struts.config.ControllerConfig] (optional)

contentType- Default content type (and optional character encoding) to be seton each response. May be overridden by the Action, JSP, or otherresource to which the request is forwarded. [text/html] (optional)

forwardPattern- Replacement pattern defining how the "path" attribute ofa <forward>element is mapped to a context-relative URL when it starts with aslash (and when the contextRelativeproperty is false).This value may consist of any combination of the following:

$M- Replaced by the module prefix of this module.

$P- Replaced by the "path" attribute of the selected<forward>element.

$$- Causes a literal dollar sign to be rendered.

$x- (Where "x" is any character not defined above) Silentlyswallowed, reserved for future use.

Ifnot specified, the default forwardPattern is consistent with theprevious behavior of forwards. [$M$P] (optional)

inputForward- Set to trueif you want the inputattribute of <action>elements to be the name of a local or global ActionForward,which will then be used to calculate the ultimate URL. Set to falseto treat the inputparameter of <action>elements as a module-relative path to the resource to be used as theinput form. [false] (optional)

locale- Set to trueif you want a Localeobject stored in the user's session if not already present. [true](optional)

maxFileSize- The maximum size (in bytes) of a file to be accepted as a fileupload. Can be expressed as a number followed by a "K","M", or "G", which are interpreted to meankilobytes, megabytes, or gigabytes, respectively. [250M] (optional)

multipartClass- The fully qualified Java class name of the multipart requesthandler class to be used with this module.[org.apache.struts.upload.CommonsMultipartRequestHandler] (optional)

nocache- Set to trueif you want the controller to add HTTP headers for defeating cachingto every response from this module. [false] (optional)

pagePattern- Replacement pattern defining how the pageattribute of custom tags using it is mapped to a context-relativeURL of the corresponding resource. This value may consist of anycombination of the following:

$M- Replaced by the module prefix of this module.

$P- Replaced by the "path" attribute of the selected<forward>element.

$$- Causes a literal dollar sign to be rendered.

$x- (Where "x" is any character not defined above) Silentlyswallowed, reserved for future use.

Ifnot specified, the default pagePattern is consistent with theprevious behavior of URL calculation. [$M$P] (optional)

processorClass- The fully qualified Java class name of the RequestProcessorsubclass to be used with this module.[org.apache.struts.action.RequestProcessor] (optional)

tempDir- Temporary working directory to use when processing file uploads.[{the directory provided by the servlet container}]

Thisexample uses the default values for several controller parameters. Ifyou only want default behavior you can omit the controller sectionaltogether.

<controller

processorClass="org.apache.struts.action.RequestProcessor"

debug="0"

contentType="text/html"/>;

MessageResource configuration

Strutshas built in support for internationalization (I18N). You can defineone or more <message-resources>elements for your webapp; modules can define their own resourcebundles. Different bundles can be used simultaneously in yourapplication, the 'key' attribute is used to specify the desiredbundle.

className- Classname of configuration bean.[org.apache.struts.config.MessageResourcesConfig] (optional)

factory- Classname of MessageResourcesFactory.[org.apache.struts.util.PropertyMessageResourcesFactory] (optional)

key- ServletContext attribute key to store this bundle.[org.apache.struts.action.MESSAGE] (optional)

null- Set to falseto display missing resource keys in your application like'???keyname???'instead of null.[true] (optional)

parameter- Name of the resource bundle. (required)

Exampleconfiguration:

<message-resources

parameter="MyWebAppResources"

null="false"/>

Thiswould set up a message resource bundle provided in the fileMyWebAppResources.propertiesunder the default key. Missing resource keys would be displayed as'???keyname???'.

PlugInconfiguration

StrutsPlugIns are configured using the <plug-in>element within the Struts configuration file. This element has onlyone valid attribute, 'className', which is the fully qualified nameof the Java class which implements theorg.apache.struts.action.PlugIninterface.

ForPlugIns that require configuration themselves, the nested<set-property>element is available.

Thisis an example using the Tiles plugin:

<plug-inclassName="org.apache.struts.tiles.TilesPlugin" >

<set-property

property="definitions-config"

value="/WEB-INF/tiles-defs.xml"/>

</plug-in>

DataSourceconfiguration

Besidesthe objects related to defining ActionMappings, the Strutsconfiguration may contain elements that create other useful objects.

The<data-sources>section can be used to specify a collection of DataSources[javax.sql.DataSource] for the use of your application. Typically, aDataSource represents a connection pool to a database or otherpersistent store. As a convenience, the Struts DataSource manager canbe used to instantiate whatever standard pool your application mayneed. Of course, if your persistence layer provides for its ownconnections, then you do not need to specify a data-sourceselement.

SinceDataSource implementations vary in what properties need to be set,unlike other Struts configuration elements, the data-sourceelement does not pre-define a slate of properties. Instead, thegeneric set-propertyfeature is used to set whatever properties your implementation mayrequire. Typically, these settings would include:

Adriver class name

Aurl to access the driver

Adescription

Andother sundry properties.

<data-sourcetype="org.apache.commons.dbcp.BasicDataSource">

<!--... set-property elements ... -->

</data-source>

InStruts 1.2.0, the GenericDataSource has been removed, and it isrecommended that you use the Commons BasicDataSource or otherDataSource implementation instead. In practice, if you need to usethe DataSource manager, you should use whatever DataSourceimplementation works best with your container or database.

Forexamples of specifying a data-sources element and using theDataSource with an Action,

TheStruts configuration file

TheBuildingController Componentschapter covered writing the form-bean and action-mapping portions ofthe Struts configuration file. These elements usually play animportant role in the development of a Struts application. The otherelements in Struts configuration file tend to be static: you set themonce and leave them alone.

These"static" configuration elements are:

controller

message-resources

plug-in

data-sources

Controller Configuration

The<controller>element allows you to configure the ActionServlet. Many of thecontroller parameters were previously defined by servletinitialization parameters in your web.xmlfile but have been moved to this section of struts-config.xmlin order to allow different modules in the same web application to beconfigured differently. For full details on available parameters seethe struts-config\_1\_2.dtdor the list below.

bufferSize- The size (in bytes) of the input buffer used when processing fileuploads. [4096] (optional)

className- Classname of configuration bean.[org.apache.struts.config.ControllerConfig] (optional)

contentType- Default content type (and optional character encoding) to be seton each response. May be overridden by the Action, JSP, or otherresource to which the request is forwarded. [text/html] (optional)

forwardPattern- Replacement pattern defining how the "path" attribute ofa <forward>element is mapped to a context-relative URL when it starts with aslash (and when the contextRelativeproperty is false).This value may consist of any combination of the following:

$M- Replaced by the module prefix of this module.

$P- Replaced by the "path" attribute of the selected<forward>element.

$$- Causes a literal dollar sign to be rendered.

$x- (Where "x" is any character not defined above) Silentlyswallowed, reserved for future use.

Ifnot specified, the default forwardPattern is consistent with theprevious behavior of forwards. [$M$P] (optional)

inputForward- Set to trueif you want the inputattribute of <action>elements to be the name of a local or global ActionForward,which will then be used to calculate the ultimate URL. Set to falseto treat the inputparameter of <action>elements as a module-relative path to the resource to be used as theinput form. [false] (optional)

locale- Set to trueif you want a Localeobject stored in the user's session if not already present. [true](optional)

maxFileSize- The maximum size (in bytes) of a file to be accepted as a fileupload. Can be expressed as a number followed by a "K","M", or "G", which are interpreted to meankilobytes, megabytes, or gigabytes, respectively. [250M] (optional)

multipartClass- The fully qualified Java class name of the multipart requesthandler class to be used with this module.[org.apache.struts.upload.CommonsMultipartRequestHandler] (optional)

nocache- Set to trueif you want the controller to add HTTP headers for defeating cachingto every response from this module. [false] (optional)

pagePattern- Replacement pattern defining how the pageattribute of custom tags using it is mapped to a context-relativeURL of the corresponding resource. This value may consist of anycombination of the following:

$M- Replaced by the module prefix of this module.

$P- Replaced by the "path" attribute of the selected<forward>element.

$$- Causes a literal dollar sign to be rendered.

$x- (Where "x" is any character not defined above) Silentlyswallowed, reserved for future use.

Ifnot specified, the default pagePattern is consistent with theprevious behavior of URL calculation. [$M$P] (optional)

processorClass- The fully qualified Java class name of the RequestProcessorsubclass to be used with this module.[org.apache.struts.action.RequestProcessor] (optional)

tempDir- Temporary working directory to use when processing file uploads.[{the directory provided by the servlet container}]

Thisexample uses the default values for several controller parameters. Ifyou only want default behavior you can omit the controller sectionaltogether.

<controller

processorClass="org.apache.struts.action.RequestProcessor"

debug="0"

contentType="text/html"/>;

Message Resources Configuration

Strutshas built in support for internationalization (I18N). You can defineone or more <message-resources>elements for your webapp; modules can define their own resourcebundles. Different bundles can be used simultaneously in yourapplication, the 'key' attribute is used to specify the desiredbundle.

className- Classname of configuration bean.[org.apache.struts.config.MessageResourcesConfig] (optional)

factory- Classname of MessageResourcesFactory.[org.apache.struts.util.PropertyMessageResourcesFactory] (optional)

key- ServletContext attribute key to store this bundle.[org.apache.struts.action.MESSAGE] (optional)

null- Set to falseto display missing resource keys in your application like'???keyname???'instead of null.[true] (optional)

parameter- Name of the resource bundle. (required)

Exampleconfiguration:

<message-resourcesparameter="MyWebAppResources" null="false" />

Thiswould set up a message resource bundle provided in the fileMyWebAppResources.propertiesunder the default key.

Missingresource keys would be displayed as '???keyname???'.

PlugIn Configuration

StrutsPlugIns are configured using the <plug-in>element within the Struts configuration file. This element has onlyone valid attribute, 'className', which is the fully qualified nameof the Java class which implements theorg.apache.struts.action.PlugIninterface.

ForPlugIns that require configuration themselves, the nested<set-property>element is available.

Thisis an example using the Tilesplugin:

<plug-inclassName="org.apache.struts.tiles.TilesPlugin" >

<set-propertyproperty="definitions-config"value="/WEB-INF/tiles-defs.xml"/>

</plug-in>

Data Source Configuration

Besidesthe objects related to defining ActionMappings, the Strutsconfiguration may contain elements that create other useful objects.

The<data-sources>section can be used to specify a collection of DataSources[javax.sql.DataSource] for the use of your application. Typically, aDataSource represents a connection pool to a database or otherpersistent store. As a convenience, the Struts DataSource manager canbe used to instantiate whatever standard pool your application mayneed. Of course, if your persistence layer provides for its ownconnections, then you do not need to specify a data-sourceselement.

SinceDataSource implementations vary in what properties need to be set,unlike other Struts configuration elements, the data-sourceelement does not pre-define a slate of properties. Instead, thegeneric set-propertyfeature is used to set whatever properties your implementation mayrequire. Typically, these settings would include:

Adriver class name

Aurl to access the driver

Adescription

Andother sundry properties.

<data-sourcetype="org.apache.commons.dbcp.BasicDataSource">

<!--... set-property elements ... -->

</data-source>

InStruts 1.2.0, the GenericDataSource has been removed, and it isrecommended that you use the Commons BasicDataSource or otherDataSource implementation instead. In practice, if you need to usethe DataSource manager, you should use whatever DataSourceimplementation works best with your container or database.

Forexamples of specifying a data-sources element and using theDataSource with an Action, see the Accessinga Database HowTo.

Configuring your application for modules

Verylittle is required in order to start taking advantage of the Strutsmodule feature. Just go through the following steps:

Preparea config file for each module.

Informthe controller of your module.

Useactions to refer to your pages.

Module Configuration Files

Backin Struts 1.0, a few "boot-strap" options were placed inthe web.xml file, and the bulk of the configuration was done in asingle struts-config.xml file. Obviously, this wasn't ideal for ateam environment, since multiple users had to share the sameconfiguration file.

InStruts 1.1, you have two options: you can list multiplestruts-config files asa comma-delimited list, or you can subdivide a larger applicationinto modules.

Withthe advent of modules, a given module has its own configuration file.This means each team (each module would presumably be developed by asingle team) has their own configuration file, and there should be alot less contention when trying to modify it.

Informing the Controller

Instruts 1.0, you listed your configuration file as an initializationparameter to the action servlet in web.xml. This is still done in1.1, but it's augmented a little. In order to tell the Strutsmachinery about your different modules, you specify multiple configinitialization parameters, with a slight twist. You'll still use"config" to tell the action servlet about your "default"module, however, for each additional module, you will list aninitialization parameter named "config/module", wheremodule is the name of your module (this gets used when determiningwhich URIs fall under a given module, so choose somethingmeaningful!). For example:

...

<init-param>

<param-name>config</param-name>

<param-value>/WEB-INF/conf/struts-default.xml</param-value>

</init-param>

<init-param>

<param-name>config/module1</param-name>

<param-value>/WEB-INF/conf/struts-module1.xml</param-value>

</init-param>

...

Thissays I have two modules. One happens to be the "default"module, which has no "/module" in it's name, and one named"module1" (config/module1). I've told the controller it canfind their respective configurations under /WEB-INF/conf (which iswhere I put all my configuration files). Pretty simple!

(Mystruts-default.xml would be equivalent to what most folks callstruts-config.xml. I just like the symmetry of having all my Strutsmodule files being named struts-<module>.xml)

Ifyou'd like to vary where the pages for each module is stored, see theforwardPatternsetting for the Controller.

Switching Modules

Thereare two basic methods to switching from one module to another. Youcan either use a forward (global or local) and specify thecontextRelative attribute with a value of true, or you can use thebuilt-in org.apache.struts.actions.SwitchAction.

Here'san example of a global forward:

...

<struts-config>

...

<global-forwards>

<forwardname="toModuleB"

contextRelative="true"

path="/moduleB/index.do"

redirect="true"/>

...

</global-forwards>

...

</struts-config>

Youcould do the same thing with a local forward declared in anActionMapping:

...

<struts-config>

...

<action-mappings>

...

<action... >

<forwardname="success"

contextRelative="true"

path="/moduleB/index.do"

redirect="true"/>

</action>

...

</action-mappings>

...

</struts-config>

Finally,you could use org.apache.struts.actions.SwitchAction,like so:

...

<action-mappings>

<actionpath="/toModule"

type="org.apache.struts.actions.SwitchAction"/>

...

</action-mappings>

...

Now,to change to ModuleB, we would use a URI like this:

http://localhost:8080/toModule.do?prefix=/moduleB&page=/index.do

Ifyou are using the "default" module as well as "named"modules (like "/moduleB"), you can switch back to the"default" module with a URI like this:

http://localhost:8080/toModule.do?prefix=&page=/index.do

That'sall there is to it! Happy module-switching!

The Web Application Deployment Descriptor

Thefinal step in setting up the application is to configure theapplication deployment descriptor (stored in file WEB-INF/web.xml)to include all the Struts components that are required. Using thedeployment descriptor for the example application as a guide, we seethat the following entries need to be created or modified.

Configure the Action Servlet Instance

Addan entry defining the action servlet itself, along with theappropriate initialization parameters. Such an entry might look likethis:

<servlet>

<servlet-name>action</servlet-name>

<servlet-class>

org.apache.struts.action.ActionServlet

</servlet-class>

<init-param>

<param-name>config</param-name>

<param-value>

/WEB-INF/struts-config.xml

</param-value>

</init-param>

<load-on-startup>1</load-on-startup>

</servlet>

Theinitialization parameters supported by the controller servlet aredescribed below. (You can also find these details in the Javadocsfor the ActionServlet class.) Square brackets describe the defaultvalues that are assumed if you do not provide a value for thatinitialization parameter.

config- Context-relative path to the XML resource containing theconfiguration information for the default module. This may also be acomma-delimited list of configuration files. Each file is loaded inturn, and its objects are appended to the internal

datastructure. [/WEB-INF/struts-config.xml].

WARNING - If you definean object of the same name in more than one configuration file, thelast one loaded quietly wins.

config/${module}- Context-relative path to the XML resource containing theconfiguration information for the application module that will usethe specified prefix (/${module}). This can be repeated as manytimes as required for multiple application modules. (Since Struts1.1)

convertNull- Force simulation of the Struts 1.0 behavior when populating forms.If set to true, the numeric Java wrapper class types (likejava.lang.Integer)will default to null (rather than 0). (Since Struts 1.1) [false]

rulesets- Comma-delimited list of fully qualified classnames of additionalorg.apache.commons.digester.RuleSetinstances that should be added to the Digesterthat will be processing struts-config.xmlfiles. By default, only the RuleSetfor the standard configuration elements is loaded. (Since Struts1.1)

validating- Should we use a validating XML parser to process the configurationfile (strongly recommended)? [true]

WARNING- Struts will not operate correctly if you define more than one<servlet>element for a controller servlet, or a subclass of the standardcontroller servlet class. The controller servlet MUST be a webapplication wide singleton.

Configure the Action Servlet Mapping

Note:The material in this section is not specific to Struts. Theconfiguration of servlet mappings is defined in the Java ServletSpecification. This section describes the most common means ofconfiguring a Struts application.

Thereare two common approaches to defining the URLs that will be processedby the controller servlet -- prefix matching and extension matching.An appropriate mapping entry for each approach will be describedbelow.

Prefixmatching means that you want all URLs that start (after the contextpath part) with a particular value to be passed to this servlet. Suchan entry might look like this:

<servlet-mapping>

<servlet-name>action</servlet-name>

<url-pattern>/do/\*</url-pattern>

</servlet-mapping>

whichmeans that a request URI to match the /logonpath described earlier might look like this:

http://www.mycompany.com/myapplication/do/logon

where/myapplicationis the context path under which your application is deployed.

Extensionmapping, on the other hand, matches request URIs to the actionservlet based on the fact that the URI ends with a period followed bya defined set of characters. For example, the JSP processing servletis mapped to the \*.jsppattern so that it is called to process every JSP page that isrequested. To use the \*.doextension (which implies "do something"), the mapping entrywould look like this:

<servlet-mapping>

<servlet-name>action</servlet-name>

<url-pattern>\*.do</url-pattern>

</servlet-mapping>

anda request URI to match the /logonpath described earlier might look like this:

http://www.mycompany.com/myapplication/logon.do

WARNING- Struts will not operate correctly if you define more than one<servlet-mapping>element for the controller servlet.

WARNING- If you are using the new module support in Struts 1.1, you shouldbe aware that only extension mapping is supported.

Configure the Struts Tag Libraries

Next,you must add an entry defining the Struts tag library.

Thestruts-bean taglib contains tags useful in accessing beans and theirproperties, as well as defining new beans (based on these accesses)that are accessible to the remainder of the page via scriptingvariables and page scope attributes. Convenient mechanisms to createnew beans based on the value of request cookies, headers, andparameters are also provided.

Thestruts-html taglib contains tags used to create struts input forms,as well as other tags generally useful in the creation of HTML-baseduser interfaces.

Thestruts-logic taglib contains tags that are useful in managingconditional generation of output text, looping over objectcollections for repetitive generation of output text, and applicationflow management.

Thestruts-tiles taglib contains tags used for combining various viewcomponents, called "tiles", into a final composite view.

Thestruts-nested taglib is an extension of other struts taglibs thatallows the use of nested beans.

Belowis how you would define all taglibs for use within your application.In practice, you would only specify the taglibs that your applicationuses:

<taglib>

<taglib-uri>/tags/struts-bean

</taglib-uri>

<taglib-location>/WEB-INF/struts-bean.tld

</taglib-location>

</taglib>

<taglib>

<taglib-uri>/tags/struts-html

</taglib-uri>

<taglib-location>/WEB-INF/struts-html.tld

</taglib-location>

</taglib>

<taglib>

<taglib-uri>/tags/struts-logic

</taglib-uri>

<taglib-location>/WEB-INF/struts-logic.tld

</taglib-location>

</taglib>

<taglib>

<taglib-uri>/tags/struts-tiles

</taglib-uri>

<taglib-location>/WEB-INF/struts-tiles.tld

</taglib-location>

</taglib>

Thistells the JSP system where to find the tag library descriptor forthis library (in your application's WEB-INF directory, instead of outon the Internet somewhere).

Configure the Struts Tag Libraries(Servlet 2.3)

Servlet2.3 Users only: The Servlet 2.3 specification simplifies thedeployment and configuration of tag libraries. The instructions abovewill work on older containers as well as 2.3 containers (Struts onlyrequires a servlet 2.2 container); however, if you're using a 2.3container such as Tomcat 4.x, you can take advantage of a simplifieddeployment.

Allthat's required to install the struts tag libraries is to copystruts.jar into your /WEB-INF/lib directory and reference the tags inyour code like this:

<%@taglib uri=http://struts.apache.org/tags-htmlprefix="html" %>

Notethat you must use the full uri defined in the various struts tlds sothat the container knows where to find the tag's class files. Youdon't have to alter your web.xml file or copy tlds into anyapplication directories.

Add Struts Components To Your Application

Touse Struts, you must copy the .tld files that you require into yourWEB-INFdirectory, and copy struts.jar(and all of the commons-\*.jarfiles) into your WEB-INF/libdirectory.

Struts Bean Tags

Thistag library contains tags useful in accessing beans and theirproperties, as well as defining new beans (based on these accesses)that are accessible to the remainder of the page via scriptingvariables and page scope attributes. Convenient mechanisms to createnew beans based on the value of request cookies, headers, andparameters are also provided.

Manyof the tags in this tag library will throw a JspExceptionat runtime when they are utilized incorrectly (such as when youspecify an invalid combination of tag attributes). JSP allows you todeclare an "error page" in the <%@page %>directive. If you wish to process the actual exception that causedthe problem, it is passed to the error page as a request attributeunder key org.apache.struts.action.EXCEPTION.

Ifyou are viewing this page from within the Struts DocumentationApplication (or online at http://struts.apache.org/),you can learn more about using these tags in the BeanTags Developer's Guide.

TagName Description

cookie Define a scripting variable based on the value(s) of thespecified request cookie.

define Define a scripting variable based on the value(s) of thespecified bean property.

headerLoad the response from a dynamic application request and make itavailable as a bean

include Render an internationalized message string to the response.

message Expose a specified item from the page context as a bean.

page Define a scripting variable based on the value(s) of the specifiedrequest parameter.

parameter Load a web application resource and make it available as a bean.

resource Define a bean containing the number of elements in a Collectionor Map.

size Expose a named Struts internal configuration object as abean.

struts Render the value of the specified bean property to thecurrent JspWriter.

Struts

Thecore of the Struts framework is a flexible control layer based onstandard technologies like Java Servlets, JavaBeans, ResourceBundles,and XML, as well as various Jakarta Commons packages. Strutsencourages application architectures based on the Model 2 approach, avariation of the classic Model-View-Controller (MVC) design paradigm.

Strutsprovides its own Controller component and integrates with othertechnologies to provide the Model and the View. For the Model, Strutscan interact with standard data access technologies, like JDBC andEJB, as well as most any third-party packages, like Hibernate,iBATIS, or Object Relational Bridge. For the View, Struts works wellwith JavaServer Pages, including JSTL and JSF, as well as VelocityTemplates, XSLT, and other presentation systems.

TheStruts framework provides the invisible underpinnings everyprofessional web application needs to survive. Struts helps youcreate an extensible development environment for your application,based on published standards and proven design patterns.

Whatis the difference between Struts 1.0 and Struts 1.1

Thenew features added to Struts 1.1 are 1. RequestProcessor class 2.Method perform()replaced by execute()in Struts base Action Class 3. Changes to web.xml andstruts-config.xml 4.Declarative exception handling 5.DynamicActionForms 6.Plug-ins 7.Multiple Application Modules 8.Nested Tags9.The Struts Validator 10.Change to the ORO package 11.Change toCommons logging 12.Removal of Admin actions 13. Deprecation of theGenericDataSource

ExplainStruts navigation flow

Aclient requests a path that matches the Action URI pattern. Thecontainer passes the request to the ActionServlet. If this is amodular application, the ActionServlet selects the appropriatemodule. The ActionServlet looks up the mapping for the path. If themapping specifies a form bean, the ActionServlet sees if there is onealready or creates one. If a form bean is in play, the ActionServletresets and populates it from the HTTP request. If the mapping has thevalidate property set to true, it calls validate on the form bean. Ifit fails, the servlet forwards to the path specified by the inputproperty and this control flow ends. If the mapping specifies anAction type, it is reused if it already exists or instantiated.

TheActionâ€™s perform or execute method is called and passed theinstantiated form bean (or null). The Action may populate the formbean, call business objects, and do whatever else is needed. TheAction returns an ActionForward to the ActionServlet. If theActionForward is to another Action URI, we begin again; otherwise,itâ€™s off to a display page or some other resource. Most often, itis a JSP, in which case Jasper, or the equivalent (not Struts),renders the page.

Whatis the difference between ActionForm and DynaActionForm

Instruts 1.0, action form is used to populate the html tags in jspusing struts custom tag.when the java code changes, the change inaction class is needed. To avoid the chages in struts 1.1 dyna actionform is introduced.This can be used to develop using xml.The dynaaction form bloats up with the struts-config.xml based definetion.

Whatis DispatchAction

TheDispatchAction class is used to group related actions into one class.DispatchAction is an abstract class, so you must override it to useit. It extends the Action class.

Itshould be noted that you dont have to use the DispatchAction to groupmultiple actions into one Action class.

Youcould just use a hidden field that you inspect to delegate tomember() methods inside of your action.

Howto call ejb from Struts

usethe Service Locator patter to look up the ejbs

OrYou can use InitialContext and get the home interface.

Whatare the various Struts tag libraries

struts-htmltag library- used for creating dynamic HTML user interfaces and forms.struts-beantag library- provides substantial enhancements to the basic capability providedby .

struts-logictag library- can manage conditional generation of output text, looping overobject collections for repetitive generation of output text, andapplication flow management.

struts-templatetag library- contains tags that are useful in creating dynamic JSP templates forpages which share a common format.

Whatis the difference between ActionErrors and ActionMessages

Thedifference between the classes is zero -- all behavior inActionErrors was pushed up into ActionMessages and all behavior inActionError was pushed up into ActionMessage. This was done in theattempt to clearly signal that these classes can be used to pass anykind of messages from the controller to the view -- errors being onlyone kind of message

Howyou will handle errors and exceptions using Struts

Thereare various ways to handle exception:

1)To handle errors server side validation can be used usingActionErrors classes can be used.

2)The exceptions can be wrapped across different layers to show a usershowable exception.

3)usingvalidators

Howyou will save the data across different pages for a particular clientrequest using Struts

Severalways. The similar to the ways session tracking is enabled. Usingcookies, URL-rewriting, SSLSession, and possibilty threw in thedatabase.

Whatwe will define in Struts-config.xml file. And explain their purpose

Themain control file in the Struts framework is the struts-config.xmlXML file, where action mappings are specified. This file's structureis described by the struts-config DTD file, which is defined athttp://jakarta.apache.org/struts/. A copy of the DTD can be found onthe /docs/dtds subdirectory of the framework's installation rootdirectory. The top-level element is struts-config. Basically, itconsists of the following elements:

data-sourcesâ€”Aset of data-source elements, describing parameters needed toinstantiate JDBC 2.0 Standard Extension DataSource objects

form-beansâ€”Aset of form-bean elements that describe the form beans that thisapplication uses

global-forwardsâ€”Aset of forward elements describing general available forward URIs

action-mappingsâ€”Aset of action elements describing a request-to-action mapping

Whatis the purpose of tiles-def.xml file, resourcebundle.properties file,validation.xml file

TheTiles Framework is an advanced version of that comes bundled with theStruts Webapp framework. Its purpose is reduce the duplicationbetween jsp pages as well as make layouts flexible and easy tomaintain. It integrates with Struts using the concept of named viewsor definitions.

Whatis Action Class. What are the methods in Action class

Actionclass is request handler in Struts. we will extend the Action classand over ride the execute() method in which we will specify thebusiness logic to be performed.

Explainabout token feature in Struts

Tokensare used to check for invalid path for by the uer:

1)if the user presses back button and submits the same page

2)orif the user refreshes the page which will result to the resubmit ofthe previous action and might lead to unstabality..

tosolve the abv probs we use tokens

1)in previous action type saveTokens(HttpServletreuest)

2)in current action check for duplication bu

if(!isValidToken())

Whatpart of MVC does Struts represent

Badquestion. Struts is a framework which supports the MVC pattern.

Whatare the core classes of struts?

Thecore classes of struts are ActionForm, Action, ActionMapping,ActionForward etc.

Whatare the Important Components of Struts?

1.Action Servlet

2.Action Classes

3.Action Form

4.Validator Framework

5.Message Resources

6.Struts Configuration XML Files

7.View components like JSP

Whatis Struts?

Strutsis a web page development framework and an open source software thathelps developers build web applications quickly and easily. Strutscombines Java Servlets, Java Server Pages, custom tags, and messageresources into a unified framework. It is a cooperative, synergisticplatform, suitable for development teams, independent developers, andeveryone between.

Howis the MVC design pattern used in Struts framework?

Inthe MVC design pattern, application flow is mediated by a centralController. The Controller delegates requests to an appropriatehandler. The handlers are tied to a Model, and each handler acts asan adapter between the request and the Model. The Model represents,or encapsulates, an application's business logic or state. Control isusually then forwarded back through the Controller to the appropriateView. The forwarding can be determined by consulting a set ofmappings, usually loaded from a database or configuration file. Thisprovides a loose coupling between the View and Model, which can makean application significantly easier to create and maintain.

Controller--Servletcontroller which supplied by Struts itself; View --- what you can seeon the screen, a JSP page and presentation components; Model ---System state and a business logic JavaBeans.

Whomakes the Struts?

Strutsis hosted by the Apache Software Foundation(ASF) as part of itsJakarta project, like Tomcat, Ant and Velocity.

Whyit called Struts?

Becausethe designers want to remind us of the invisible underpinnings thathold up our houses, buildings, bridges, and ourselves when we are onstilts. This excellent description of Struts reflect the role theStruts plays in developing web applications.

Dowe need to pay the Struts if being used in commercial purpose?

No.Struts is available for commercial use at no charge under the ApacheSoftware License. You can also integrate the Struts components intoyour own framework just as if they were writtern in house without anyred tape, fees, or other hassles

Whatare the core classes of Struts?

Action,ActionForm, ActionServlet, ActionMapping, ActionForward are basicclasses of Structs.

Whatis the design role played by Struts?

Therole played by Structs is controller in Model/View/Controller(MVC)style. The View is played by JSP and Model is played by JDBC orgeneric data source classes. The Struts controller is a set ofprogrammable components that allow developers to define exactly howthe application interacts with the user.

HowStruts control data flow?

Strutsimplements the MVC/Layers pattern through the use of ActionForwardsand ActionMappings to keep control-flow decisions out of presentationlayer.

Whatconfiguration files are used in Struts?

--ApplicationResourcesl.properties

--struts-config.xml

Thesetwo files are used to bridge the gap between the Controller and theModel.

What helpers in the form of JSP pages areprovided in Struts framework?

--struts-html.tld

--struts-bean.tld

--struts-logic.tld

Is Struts efficient?

--TheStruts is not only thread-safe but thread-dependent(instantiates eachAction once and allows other requests to be threaded through theoriginal object.

--ActionFormbeans minimize subclass code and shorten subclass hierarchies

--TheStruts tag libraries provide general-purpose functionality

--TheStruts components are reusable by the application

--TheStruts localization strategies reduce the need for redundant JSPs

--TheStruts is designed with an open architecture--subclass available

--TheStruts is lightweight (5 core packages, 5 tag libraries)

--TheStruts is open source and well documented (code to be examinedeasily)

--TheStruts is model neutral

Whatis Jakarta Struts Framework? - JakartaStruts is open source implementation of MVC (Model-View-Controller)pattern for the development of web based applications. Jakarta Strutsis robust architecture and can be used for the development ofapplication of any size. Struts framework makes it much easier todesign scalable, reliable Web applications with Java.

Whatis ActionServlet?

Theclass org.apache.struts.action.ActionServletis the called the ActionServlet. In the the Jakarta Struts Frameworkthis class plays the role of controller. All the requests to theserver goes through the controller. Controller is responsible forhandling all the requests.

Howyou will make available any Message Resources Definitions file to theStruts Framework Environment?

MessageResources Definitions file are simple .properties files and thesefiles contains the messages that can be used in the struts project.Message Resources Definitions files can be added to thestruts-config.xml file through <message-resources /> tag.

Example:

<message-resourcesparameter=â€MessageResourcesâ€ />

Whatis Action Class?

The Action Class ispart of the Model and is a wrapper around the business logic. Thepurpose of Action Class is to translate the HttpServletRequest to thebusiness logic. To use the Action, we need to Subclass andoverwrite the execute() method. In the Action Class all thedatabase/business processing are done. It is advisable to perform allthe database related stuffs in the Action Class. The ActionServlet(commad) passes the parameterized class to Action Form using theexecute() method. The return type of the execute method isActionForward which is used by the Struts Framework to forward therequest to the file as per the value of the returned ActionForwardobject.

Writecode of any Action Class?

Hereis the code of Action Class that returns the ActionForward object.

importjavax.servlet.http.HttpServletRequest;

importjavax.servlet.http.HttpServletResponse;

importorg.apache.struts.action.Action;

importorg.apache.struts.action.ActionForm;

importorg.apache.struts.action.ActionForward;

importorg.apache.struts.action.ActionMapping;

publicclass TestAction extends Action

{

publicActionForward execute(

ActionMappingmapping,

ActionFormform,

HttpServletRequestrequest,

HttpServletResponseresponse) throws Exception

{

return mapping.findForward(\"testAction\");

}

}

Whatis ActionForm?

AnActionForm is a JavaBean that extendsorg.apache.struts.action.ActionForm.ActionForm maintains the session state for web application and theActionForm object is automatically populated on the server side withdata entered from a form on the client side.

Whatis Struts Validator Framework?

StrutsFramework provides the functionality to validate the form data. Itcan be use to validate the data on the users browser as well as onthe server side. Struts Framework emits the java scripts and it canbe used validate the form data on the client browser. Server sidevalidation of form can be accomplished by sub classing your From Beanwith DynaValidatorForm class. The Validator framework was developedby David Winterfeldt as third-party add-on to Struts. Now theValidator framework is a part of Jakarta Commons project and it canbe used with or without Struts. The Validator framework comesintegrated with the Struts Framework and can be used without doingany extra settings.

Givethe Details of XML files used in Validator Framework?

TheValidator Framework uses two XML configuration filesvalidator-rules.xml and validation.xml. The validator-rules.xmldefines the standard validation routines, these are reusable and usedin validation.xml. to define the form specific validations. Thevalidation.xml defines the validations applied to a form bean.

Howyou will display validation fail errors on jsp page? Thefollowing tag displays all the errors:

<html:errors/>

How you will enable front-end validation based on the xml invalidation.xml? The <html:javascript>tag to allow front-end validation based on the xml in validation.xml.For example the code: <html:javascriptformName=â€logonFormâ€ dynamicJavascript=â€trueâ€staticJavascript=â€trueâ€ />generates the client side java script for the form â€œlogonFormâ€ asdefined in the validation.xml file. The <html:javascript>when added in the jsp file generates the client site validationscript.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_€€€\_\_\_\_\_\_\_\_\_\_€€€\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

====================spring===================

1. What is IOC (or Dependency Injection)?

Thebasic concept of the Inversion of Control pattern (also known asdependency injection) is that you do not create your objects butdescribe how they should be created. You don't directly connect yourcomponents and services together in code but describe which servicesare needed by which components in a configuration file. A container(in the case of the Spring framework, the IOC container) is thenresponsible for hooking it all up.

i.e., Applying IoC, objectsare given their dependencies at creation time by some external entitythat coordinates each object in the system. That is, dependencies areinjected into objects. So, IoC means an inversion of responsibilitywith regard to how an object obtains references to collaboratingobjects.

2. Whatare the different types of IOC (dependency injection) ?

There are threetypes of dependency injection:

ConstructorInjection (e.g. Pico container, Spring etc): Dependencies areprovided as constructor parameters.

SetterInjection (e.g. Spring): Dependencies are assigned throughJavaBeans properties (ex: setter methods).

InterfaceInjection (e.g. Avalon): Injection is done through an interface.

Note: Springsupports only Constructor and Setter Injection

3. Whatare the benefits of IOC (Dependency Injection)?

Benefits of IOC(Dependency Injection) are as follows:

Minimizes theamount of code in your application. With IOC containers you do notcare about how services are created and how you get references tothe ones you need. You can also easily add additional services byadding a new constructor or a setter method with little or no extraconfiguration.

Make your application moretestable by not requiring any singletons or JNDI lookup mechanismsin your unit test cases. IOC containers make unit testing andswitching implementations very easy by manually allowing you toinject your own objects into the object under test.

Loose coupling is promoted withminimal effort and least intrusive mechanism. The factory designpattern is more intrusive because components or services need to berequested explicitly whereas in IOC the dependency is injected intorequesting piece of code. Also some containers promote the design tointerfaces not to implementations design concept by encouragingmanaged objects to implement a well-defined service interface ofyour own.

IOC containers support eagerinstantiation and lazy loading of services. Containers also providesupport for instantiation of managed objects, cyclical dependencies,life cycles management, and dependency resolution between managedobjects etc.

4. What is Spring ?

Spring is anopen source framework created to address the complexity of enterpriseapplication development. One of the chief advantages of the Springframework is its layered architecture, which allows you to beselective about which of its components you use while also providinga cohesive framework for J2EE application development.

5. Whatare the advantages of Spring framework?

The advantagesof Spring are as follows:

Spring has layeredarchitecture. Use what you need and leave you don't need now.

Spring EnablesPOJO Programming. There is no behind the scene magic here. POJOprogramming enables continuous integration and testability.

DependencyInjection and Inversion of Control Simplifies JDBC

Open source andno vendor lock-in.

6. Whatare features of Spring ?

Lightweight:

spring is lightweight when it comes to size and transparency. Thebasic version of spring framework is around 1MB. And the processingoverhead is also very negligible.

Inversion ofcontrol (IOC):

Loose coupling is achieved in spring using the technique Inversion ofControl. The objects give their dependencies instead of creating orlooking for dependent objects.

Aspect oriented(AOP):

Spring supports Aspect oriented programming and enables cohesivedevelopment by separating application business logic from systemservices.

Container:

Spring contains and manages the life cycle and configuration ofapplication objects.

MVC Framework:

Spring comes with MVC web application framework, built on core Springfunctionality. This framework is highly configurable via strategyinterfaces, and accommodates multiple view technologies like JSP,Velocity, Tiles, iText, and POI. But other frameworks can be easilyused instead of Spring MVC Framework.

TransactionManagement:

Spring framework provides a generic abstraction layer for transactionmanagement. This allowing the developer to add the pluggabletransaction managers, and making it easy to demarcate transactionswithout dealing with low-level issues. Spring's transaction supportis not tied to J2EE environments and it can be also used in containerless environments.

JDBC ExceptionHandling:

The JDBC abstraction layer of the Spring offers a meaningfulexception hierarchy, which simplifies the error handling strategy.Integration with Hibernate, JDO, and iBATIS: Spring provides bestIntegration services with Hibernate, JDO and iBATIS

7. Howmany modules are there in Spring? What are they?

Spring comprises ofseven modules. They are..

The corecontainer:

The core container provides the essential functionality of the Springframework. A primary component of the core container is theBeanFactory,an implementation of the Factory pattern. The BeanFactoryapplies the Inversion of Control (IOC) pattern to separatean application's configuration and dependency specification from theactual application code.

Spring context:

The Spring context is a configuration file that provides contextinformation to the Spring framework. The Spring context includesenterprise services such as JNDI, EJB, e-mail, internalization,validation, and scheduling functionality.

Spring AOP:

The Spring AOP module integrates aspect-oriented programmingfunctionality directly into the Spring framework, through itsconfiguration management feature. As a result you can easilyAOP-enable any object managed by the Spring framework. The Spring AOPmodule provides transaction management services for objects in anySpring-based application. With Spring AOP you can incorporatedeclarative transaction management into your applications withoutrelying on EJB components.

Spring DAO:

The Spring JDBC DAO abstraction layer offers a meaningful exceptionhierarchy for managing the exception handling and error messagesthrown by different database vendors. The exception hierarchysimplifies error handling and greatly reduces the amount of exceptioncode you need to write, such as opening and closing connections.Spring DAO's JDBC-oriented exceptions comply to its generic DAOexception hierarchy.

Spring ORM:

The Spring framework plugs into several ORM frameworks to provide itsObject Relational tool, including JDO, Hibernate, and iBatis SQLMaps. All of these comply to Spring's generic transaction and DAOexception hierarchies.

Spring Webmodule:

The Web context module builds on top of the application contextmodule, providing contexts for Web-based applications. As a result,the Spring framework supports integration with Jakarta Struts. TheWeb module also eases the tasks of handling multi-part requests andbinding request parameters to domain objects.

Spring MVCframework:

The Model-View-Controller (MVC) framework is a full-featured MVCimplementation for building Web applications. The MVC framework ishighly configurable via strategy interfaces and accommodates numerousview technologies including JSP, Velocity, Tiles, iText, and POI.

8. Whatare the types of Dependency Injection Spring supports?

SetterInjection:

Setter-based DI is realized by calling setter methods on your beansafter invoking a no-argument constructor or no-argument staticfactory method to instantiate your bean.

ConstructorInjection:

Constructor-based DI is realized by invoking a constructor with anumber of arguments, each representing a collaborator.

9. Whatis Bean Factory ?

A BeanFactory islike a factory class that contains a collection of beans. TheBeanFactory holds Bean Definitions of multiple beans within itselfand then instantiates the bean whenever asked for by clients.

BeanFactory isable to create associations between collaborating objects as theyare instantiated. This removes the burden of configuration from beanitself and the beans client.

BeanFactoryalso takes part in the life cycle of a bean, making calls to custominitialization and destruction methods.

10. Whatis Application Context?

A bean factoryis fine to simple applications, but to take advantage of the fullpower of the Spring framework, you may want to move up to Springsmore advanced container, the application context. On the surface, anapplication context is same as a bean factory.Both load beandefinitions, wire beans together, and dispense beans upon request.But it also provides:

A means forresolving text messages, including support for internationalization.

A generic way toload file resources.

Events to beansthat are registered as listeners.

11. Whatis the difference between Bean Factory and Application Context ?

On the surface,an application context is same as a bean factory. But applicationcontext offers much more..

Applicationcontexts provide a means for resolving text messages, includingsupport for i18n of those messages.

Applicationcontexts provide a generic way to load file resources, such asimages.

Applicationcontexts can publish events to beans that are registered aslisteners.

Certain operationson the container or beans in the container, which have to be handledin a programmatic fashion with a bean factory, can be handleddeclaratively in an application context.

ResourceLoadersupport: Springâ€™s Resource interface us a flexible genericabstraction for handling low-level resources. An application contextitself is a ResourceLoader, Hence provides an application withaccess to deployment-specific Resource instances.

MessageSourcesupport: The application context implements MessageSource, an int

12. Whatare the common implementations of the Application Context ?

Thethree commonly used implementation of 'Application Context' are

ClassPathXmlApplicationContext: It Loads context definition from an XML file located in theclasspath, treating context definitions as classpath resources. Theapplication context is loaded from the application's classpath byusing the code .

ApplicationContextcontext = new ClassPathXmlApplicationContext("bean.xml");

FileSystemXmlApplicationContext: It loads context definition from an XML file in thefilesystem. The application context is loaded from the file systemby using the code .

ApplicationContextcontext = new FileSystemXmlApplicationContext("bean.xml");

XmlWebApplicationContext: It loads context definition from an XML file contained withina web application.

13. Howis a typical spring implementation look like ?

Fora typical Spring Application we need the following files:

An interfacethat defines the functions.

An Implementation that containsproperties, its setter and getter methods, functions etc.,

Spring AOP (Aspect OrientedProgramming)

A XML file called Springconfiguration file.

Client program that uses thefunction.

14. What is the typical Bean life cycle in Spring Bean Factory Container?

Beanlife cycle in Spring Bean Factory Container is as follows:

The springcontainer finds the beanâ€™s definition from the XML file andinstantiates the bean.

Using the dependency injection,spring populates all of the properties as specified in the beandefinition

If the bean implements theBeanNameAware interface, the factory calls setBeanName()passing the beanâ€™s ID.

If the bean implements theBeanFactoryAware interface, the factory calls setBeanFactory(),passing an instance of itself.

If there are anyBeanPostProcessors associated with the bean, their post-ProcessBeforeInitialization()methods will be called.

If an init-method is specifiedfor the bean, it will be called.

Finally, if there are anyBeanPostProcessors associated with the bean, theirpostProcessAfterInitialization()methods will be called.

15. Whatdo you mean by Bean wiring ?

The act ofcreating associations between application components (beans) withinthe Spring container is reffered to as Bean wiring.

16. Whatdo you mean by Auto Wiring?

TheSpring container is able to autowire relationships betweencollaborating beans. This means that it is possible to automaticallylet Spring resolve collaborators (other beans) for your bean byinspecting the contents of the BeanFactory. The autowiringfunctionality has five modes.

no

byName

byType

constructor

autodirect

17. Whatis DelegatingVariableResolver?

Springprovides a custom JavaServer Faces VariableResolver implementationthat extends the standard Java Server Faces managed beans mechanismwhich lets you use JSF and Spring together. This variable resolver iscalled as DelegatingVariableResolver

18. Howto integrate Java Server Faces (JSF) with Spring?

JSFand Spring do share some of the same features, most noticeably in thearea of IOC services. By declaring JSF managed-beans in thefaces-config.xml configuration file, you allow the FacesServlet toinstantiate that bean at startup. Your JSF pages have access to thesebeans and all of their properties.We can integrate JSF and Spring intwo ways:

DelegatingVariableResolver:Spring comes with a JSF variable resolver that lets you use JSF andSpring together.

<?xmlversion="1.0" encoding="UTF-8"?>

<!DOCTYPEbeans PUBLIC "-//SPRING//DTD BEAN//EN"

"http://www.springframework.org/dtd/spring-beans.dtd">

<faces-config>

<application>

<variable-resolver>

org.springframework.web.jsf.DelegatingVariableResolver

</variable-resolver>

</application>

</faces-config>

The DelegatingVariableResolver will first delegate value lookups tothe default resolver of the underlying JSF implementation, and thento Spring's 'business context' WebApplicationContext. This allows oneto easily inject dependencies into one's JSF-managed beans.

FacesContextUtils:customVariableResolver works well when mapping one's properties to beansin faces-config.xml, but at times one may need to grab a beanexplicitly. The FacesContextUtils class makes this easy. It issimilar to WebApplicationContextUtils, except that it takes aFacesContext parameter rather than a ServletContext parameter.

ApplicationContextctx =FacesContextUtils.getWebApplicationContext(FacesContext.getCurrentInstance());

19. Whatis Java Server Faces (JSF) - Spring integration mechanism?

Spring providesa custom JavaServer Faces VariableResolver implementation thatextends the standard JavaServer Faces managed beans mechanism. Whenasked to resolve a variable name, the following algorithm isperformed:

Does a bean withthe specified name already exist in some scope (request, session,application)? If so, return it

Is there astandard JavaServer Faces managed bean definition for this variablename? If so, invoke it in the usual way, and return the bean thatwas created.

Is thereconfiguration information for this variable name in the SpringWebApplicationContext for this application? If so, use it to createand configure an instance, and return that instance to the caller.

If there is nomanaged bean or Spring definition for this variable name, returnnull instead.

BeanFactoryalso takes part in the life cycle of a bean, making calls to custominitialization and destruction methods.

As a result of this algorithm, you can transparently use eitherJavaServer Faces or Spring facilities to create beans on demand.

20. Whatis Significance of JSF- Spring integration ?

Spring - JSFintegration is useful when an event handler wishes to explicitlyinvoke the bean factory to create beans on demand, such as a beanthat encapsulates the business logic to be performed when a submitbutton is pressed.

21. Howto integrate your Struts application with Spring?

To integrateyour Struts application with Spring, we have two options:

Configure Springto manage your Actions as beans, using the ContextLoaderPlugin, andset their dependencies in a Spring context file.

SubclassSpring's ActionSupport classes and grab your Spring-managed beansexplicitly using a getWebApplicationContext()method

22. Whatare ORMâ€™s Spring supports ?

Springsupports the following ORMâ€™s :

Hibernate

iBatis

JPA (JavaPersistence API)

TopLink

JDO (Java DataObjects)

OJB

23. Howto integrate Spring and Hibernate ?

Springand Hibernate can integrate using Springâ€™s SessionFactory calledLocalSessionFactory. The integration process is of 3 steps.

ConfigureHibernate mappings.

ConfigureHibernate properties.

Wire dependantobject to SessionFactory.

Moreabout Spring-Hibernate integration....

24. What are the ways to access Hibernate using Spring ?

Thereare two ways to access Hibernate from Spring:

ThroughHibernate Template.

Subclassing HibernateDaoSupport

Extending HibernateDaoSupportand Applying an AOP Interceptor

25. Whatare Bean scopes in Spring Framework ?

TheSpring Framework supports exactly five scopes (of which three areavailable only if you are using a web-aware ApplicationContext). Thescopes supported are listed below:

Scope

Description

singleton

Scopes a single bean definition toa single object instance per Spring IoC container.

prototype

Scopes a single bean definition toany number of object instances.

request

Scopes a single bean definition tothe lifecycle of a single HTTP request; that is each and everyHTTP request will have its own instance of a bean created off theback of a single bean definition. Only valid in the context of aweb-aware Spring ApplicationContext.

session

Scopes a single bean definition tothe lifecycle of a HTTP Session.Only valid in the context of a web-aware SpringApplicationContext.

globalsession

Scopes a single bean definition tothe lifecycle of a global HTTP Session.Typically only valid when used in a portlet context. Only validin the context of a web-aware Spring ApplicationContext.

26. Whatis AOP?

Aspect-orientedprogramming, or AOP, is a programming technique that allowsprogrammers to modularize crosscutting concerns, or behavior thatcuts across the typical divisions of responsibility, such as loggingand transaction management. The core construct of AOP is the aspect,which encapsulates behaviors affecting multiple classes into reusablemodules.

27. Howthe AOP used in Spring?

AOPis used in the Spring Framework:To provide declarativeenterprise services, especially as a replacement for EJB declarativeservices. The most important such service is declarative transactionmanagement, which builds on the Spring Framework's transactionabstraction.To allow users to implement custom aspects, complementingtheir use of OOP with AOP.

28. Whatdo you mean by Aspect ?

Amodularization of a concern that cuts across multiple objects.Transaction management is a good example of a crosscutting concern inJ2EE applications. In Spring AOP, aspects are implemented usingregular classes (the schema-based approach) or regular classesannotated with the @Aspect annotation (@AspectJ style).

29. Whatdo you mean by JointPoint?

A point duringthe execution of a program, such as the execution of a method or thehandling of an exception. In Spring AOP, a join point alwaysrepresents a method execution.

30. Whatdo you mean by Advice?

Action taken byan aspect at a particular join point. Different types of adviceinclude "around," "before" and "after"advice. Many AOP frameworks, including Spring, model an advice as aninterceptor, maintaining a chain of interceptors "around"the join point.

31. Whatare the types of Advice?

Types of advice:

Before advice:Advice that executes before a join point, but which does not havethe ability to prevent execution flow proceeding to the join point(unless it throws an exception).

Afterreturning advice: Advice to be executed after a join pointcompletes normally: for example, if a method returns withoutthrowing an exception.

After throwingadvice: Advice to be executed if a method exits by throwing anexception.

After(finally) advice: Advice to be executed regardless of the meansby which a join point exits (normal or exceptional return).

Aroundadvice: Advice that surrounds a join point such as a methodinvocation. This is the most powerful kind of advice. Around advicecan perform custom behavior before and after the method invocation.It is also responsible for choosing whether to proceed to the joinpoint or to shortcut the advised method execution by returning itsown return value or throwing an exception

32. Whatare the types of the transaction management Spring supports ?

SpringFramework supports:

Programmatictransaction management.

Declarativetransaction management.

33. Whatare the benefits of the Spring Framework transaction management ?

TheSpring Framework provides a consistent abstraction for transactionmanagement that delivers the following benefits:

Provides aconsistent programming model across different transaction APIs suchas JTA, JDBC, Hibernate, JPA, and JDO.

Supportsdeclarative transaction management.

Provides a simplerAPI for programmatic transaction management than a number of complextransaction APIs such as JTA.

Integrates verywell with Spring's various data access abstractions.

34. Why most users of the Spring Framework choose declarative transactionmanagement ?

Mostusers of the Spring Framework choose declarative transactionmanagement because it is the option with the least impact onapplication code, and hence is most consistent with the ideals of anon-invasive lightweight container.

35. Explainthe similarities and differences between EJB CMT and the SpringFramework's declarative transaction

management ?

Thebasic approach is similar: it is possible to specify transactionbehavior (or lack of it) down to individual method level. It is

possible to make a setRollbackOnly() call within a transactioncontext if necessary. The differences are:

Unlike EJB CMT,which is tied to JTA, the Spring Framework's declarative transactionmanagement works in any environment. It can work with JDBC, JDO,Hibernate or other transactions under the covers, with configurationchanges only.

The SpringFramework enables declarative transaction management to be appliedto any class, not merely special classes such as EJBs.

The SpringFramework offers declarative rollback rules: this is a feature withno EJB equivalent. Both programmatic and declarative support forrollback rules is provided.

The SpringFramework gives you an opportunity to customize transactionalbehavior, using AOP. With EJB CMT, you have no way to influence thecontainer's transaction management other than setRollbackOnly().

The SpringFramework does not support propagation of transaction contextsacross remote calls, as do high-end application servers.

37. Whento use programmatic and declarative transaction management ?

Programmatictransaction management is usually a good idea only if you have asmall number of transactional operations.

On the other hand, ifyour application has numerous transactional operations, declarativetransaction management is usually worthwhile. It keeps transactionmanagement out of business logic, and is not difficult to configure.

38. Explainabout the Spring DAO support ?

TheData Access Object (DAO) support in Spring is aimed at making it easyto work with data access technologies like JDBC, Hibernate or JDO ina consistent way. This allows one to switch between the persistencetechnologies fairly easily and it also allows one to code withoutworrying about catching exceptions that are specific to eachtechnology.

39. Whatare the exceptions thrown by the Spring DAO classes ?

Spring DAOclasses throw exceptions which are subclasses ofDataAccessException(org.springframework.dao.DataAccessException).Springprovides a convenient translation from technology-specific exceptionslike SQLExceptionto its own exception class hierarchy with the DataAccessExceptionas the root exception. These exceptions wrap the original exception.

40. Whatis SQLExceptionTranslator ?

SQLExceptionTranslator,is an interface to be implemented by classes that can translatebetween SQLExceptions and Spring's own data-access-strategy-agnosticorg.springframework.dao.DataAccessException.

41. Whatis Spring's JdbcTemplate ?

Spring'sJdbcTemplate is central class to interact with a databasethrough JDBC. JdbcTemplate provides many convenience methods fordoing things such as converting database data into primitives orobjects, executing prepared and callable statements, and providingcustom database error handling.

JdbcTemplatetemplate = new JdbcTemplate(myDataSource);

42. Whatis PreparedStatementCreator ?

PreparedStatementCreator:

Is one of the mostcommon used interfaces for writing data to database.

Has one method â€“createPreparedStatement(Connection)

Responsible forcreating a PreparedStatement.

Does not needto handle SQLExceptions.

43. Whatis SQLProvider ?

SQLProvider:

Has one method â€“getSql()

Typicallyimplemented by PreparedStatementCreator implementers.

Useful fordebugging.

44. Whatis RowCallbackHandler ?

The RowCallbackHandler interfaceextracts values from each row of a ResultSet.

Has one method â€“processRow(ResultSet)

Called for eachrow in ResultSet.

Typicallystateful