Java™ Technology Concept Map

What is Java Technology?
This diagram is a model of Java™ technology. The diagram explains Java technology by placing it in the context of related concepts and examples, and by defining its major components and the connections between them. It shows how developers use Java technology to create programs that benefit people everywhere, and explains how computers and networks relate to Java technology.

The diagram is intended to help developers who are familiar with one part of the Java platform understand other parts. It relates unfamiliar technologies to ones with which developers may already be familiar. The diagram also provides an overview for developers who are new to Java technology and an introduction for non-programmers who want to improve their ability to converse with developers. For more information, visit the web site at http://java.sun.com.

Concept Maps
The diagram takes the form of a concept map – a web of linked terms showing both overall structure and details. By showing everything – the forest and the trees – in a single view, concept maps help people visualize mental models and clarify thoughts.

In concept maps, verbs connect nouns to form propositions. Examples and details accompany the terms. More important terms receive visual emphasis; less important terms and examples are in gray. Purple terms and purple lines indicate a process. Terms followed by a number link to terms preceded by the same number.

Java — which is a logo — is distinguished by features and benefits — such as object-oriented programming language ▸ 9, explicit error handling ▸ 22, garbage collection ▸ 21, security model ▸ 20, architecturally independent applications ▸ 19, Java virtual machine ▸ 18

Java — is a trademark

Java — is a name for a ... programming language ▸ 9; is also a name for ... SDKs and JREs ▸ 10; defines a set of ... APIs ▸ 13; is implemented as a set of ... platforms ▸ 27; can be used to write ... programs ▸ 24

Downloads and documentation are available at http://java.sun.com.

Selected Java products have been localized into many languages. This localization is one of the key factors for adoption of Java technology worldwide.

learn and use — **Developers** — to create and run — **programs** — that make — **devices** — and the — **internet** — useful for — **people**

Developers may submit or comment on a ... Java Specification Request ▸ 3; who are members of the Java Community Process serve on ... expert groups ▸ 5; form ... Java developer communities ▸ 4; comment on a ... public draft ▸ 6; participate in the ... Java Community Process ▸ 2 ▸ ▸; may request revisions from the ... maintenance lead ▸ 7; who are members of the JCP elect the ... executive committee ▸ 8; often use ... development environments ▸ 16; participate in the ... development process ▸ ▸

Java Community Process℠ — begins with ... Java forums — often discuss Java in ... Java developer communities — members join the JCP by signing the ... logo ▸ 1

Sun works with ... companies — supports the development of ... Java ▸ 0; provides ... documentation ▸ 22; owns the ... Java trademark ▸ 1; make(s) ... SDKs ▸ 20; makes versions of a ... JVM ▸ 18; provides ... — is represented on ... Sun Microsystems — are represented on ... Java Community Process℠ (JCP) — includes members from ... Companies include IBM, Motorola, Nokia, Oracle, Nowell, Netscape, Hewlett-Packard, Apple, and BEA.

Java Community Process℠ (JCP) — oversees by the ... Executive Committee — comments on ... — develops ... Java Specification Request (JSR) ▸ 3

Java Specification Request (JSR) — if accepted becomes a ... community draft — may become a ... public draft — may become a ... new Java specification — is instantiated in a ... Reference Implementation & Technology Compatibility Kit (TCK)

A proposal to develop a new specification or significant revision to an existing spec.

A Java specification describes some aspect of Java technology, including language, virtual machines, platform editions, profiles, and APIs.

A Reference Implementation is a working example of the JSR. Other companies will produce their own implementations. The TCK is a suite of tests, tools, and documentation that provides a standard way of testing an implementation for compliance with a Java specification.

Program Management Office — administers the ... chairs the ... exists within ... Executive Committee

specification leads (spec leads) — include ... expert groups — consist of members of and do the work of the ... Java Community Process℠ (JCP)

expert groups — develop ... Java Specification Request (JSR) ▸ 3

maintenance lead — may request major revisions via ... may revise ... public draft ▸ 6

JSPA — The Java Specification Participation Agreement is a one-year renewable agreement that allows signatories to become members of the JCP. Alternatively, developers can sign the more limited Individual Expert Participation Agreement (IEPA).

alternate implementations — Alternate implementations are written to the same specification and pass the TCK, but can be licensed differently and may have enhanced features beyond the reference implementation.

Programming language — is defined by the ... Java Language Specification ▸ 43; is used to write ... programs ▸ 24; is used to write ... class libraries ▸ 10

The Java language has roots in C, Objective C, Smalltalk, and LISP.

object-oriented programming language ▸ 9 — is based on the ... syntax and keywords — defines ... Java Language Specification ▸ 43

As opposed to a purely procedural programming language.

principles — provide — benefits and advantages

Such as the object model, inheritance, modularity, abstraction, hierarchy, implementation hiding, information hiding, encapsulation, and polymorphism.

benefits and advantages:
- reduced development time
- ability to write higher-quality code
- more maintainable code
- ability to call non-Java functions

Java Native Interface (JNI) provides a method for calling native functions such as operating system or legacy library functions.

Java object model

Class libraries are organized collections of prebuilt classes and functions used to create other classes. Class libraries can be part of the Java standard – meaning they have been ratified by the Java Community Process – or created by individual developers for their own or their company's use.

An application programming interface is the written or understood specification of how a piece of software interacts with the outside world. It specifies what input the application or application component accepts, how the software will behave, and what output it will provide. All computer software relies on a variety of APIs to achieve tasks such as writing a file or displaying text. APIs often depend on other APIs.

Particular to Java, interfaces are source code files that define a set of functions (methods) that are required by a class. Interfaces name methods and set an expectation for each method's behavior, but do not implement the methods. Interfaces allow classes to be written in such a way that the code can work with any type of class that supports the interface.

Abstract classes permit child classes to inherit a defined method or to create groups of related classes for polymorphic behavior.

class libraries — implement — APIs — define — interfaces

packages — implement — interfaces; may include — classes; may include — class libraries; are standard in — J2EE, J2SE, J2ME

objects — may implement — interfaces; are runtime instances of ... classes ▸ 12; have — methods and fields; are created by — constructors

Runtime instances of classes.

serialization — In runtime, refers to the process of turning an object into a data stream for purposes of transmission or storage.

Classes are source code files that describe a unit of programming. A class relates a set of data (fields) and functions (methods) that use the data.

classes — implement — interfaces; are standard in — J2EE, J2SE, J2ME; running in RAM are — objects; may be categorized as — class libraries; are constrained by — scope; are distinguished by — methods and fields

superclass (parent class) — may be an — abstract class

subclass (child class) — inherits methods and properties from a — may also be a — superclass (parent class); may override — methods and fields; may add new methods and fields

methods and fields — are contained in — classes; are used by — constructors; are used by — objects; are used by — interfaces; are used by — scope

scope — Scope limits access to methods and properties. Levels include:
- public: can be called or manipulated freely
- protected: can be called or manipulated by a class in the same package
- private: can be called or manipulated only by the same class.

types — field types can be — primitives or — object references — refer to ... objects ▸ 55

field or class types can be ... classes ▸ 12

Primitives generally enable basic math or comparison operations.

inner classes — Methods are detailed instructions that cause a running object to take action or manipulate data.

methods — are contained in — classes; are invoked by — constructors; are invoked by — objects; are invoked by — interfaces ▸ 14; are invoked by — constructors; may define "return" — types

constructors — are invoked by — methods; are used to create — objects ▸ 15; are used to create — interfaces ▸ 14

fields — define a location to store — data; are contained in — classes; have — types

Fields are sometimes referred to as properties.

data — refer to ... objects ▸ 55

statements — In running programs, statements prescribe actions or a sequence of actions.

declarations — In running programs, declarations define a variable's existence and determine scope.

comments — Notes that developers include to explain source code to themselves and others.

Javadoc™ comments — are processed by the — Javadoc tool — to create — browsable documentation — serves as the — API spec — defines the syntax of ... Java ▸ 13

Comments specifically formatted to be processed by the Javadoc tool.

Application programming interface specification.

Java virtual machine (JVM™) — is defined by the Java virtual machine specification

Architecturally dependent machines that are available for nearly every contemporary operating system.

enables — architecturally independent applications — enables — portability

One application can run in multiple environments without being rewritten or recompiled.

enforces a — security model — controls access to — critical resources — including memory and hard drive.

improves — performance — can be improved by — Just-In-Time compilation — significantly improves virtual machine performance by translating groups of instructions from byte code to machine code rather than one instruction at a time.

can be improved by — dynamic compilation — Goes beyond Just-in-Time compilation by examining the code as it runs and focusing optimization efforts on the most critical pieces of code as it executes. Dynamic compilation is enacted by the Java HotSpot™ virtual machine.

implements — garbage collection — Watching for data in memory that is no longer needed and freeing the memory for reuse. Other programming languages require programmers to free memory explicitly when a piece of data is no longer needed, which can increase the development effort.

implements — explicit error handling — The virtual machine may encounter a problem, or exception, while a program is running. The Java language requires that pieces of code which are likely to encounter a particular error offer a mechanism for handling the exception at runtime. When an error is encountered, the virtual machine executes the code designed to respond to the error.

Development process

development environments — may include — tools — such as ... are used to write, test, and debug

Java developers often use specific development environments called Integrated Development Environments, or IDEs.

visual programming tools — define — class structures; define — user interface frameworks; are used to build — source code framework — is edited in the ... is the foundation for the completed ... — is compiled by — compiler — produces — byte code — runs on — virtual machine — interprets — byte code — runs on — machine instructions — are processed by — microprocessor — runs — fragment or application — may be a — Java applet — runs on — Java-enabled browser — runs on — Java virtual machine — runs on — operating system — runs on — microprocessor

Class structures are basic source code that provides a framework for the classes, describes the class hierarchy, and provides the names of the class methods and properties. Does not include implementation details (method definitions).

class structures — are contained in — classes; are used to build — source code framework

The combination of the class structures and the user interface framework into a skeleton of the application's overall source code. Does not include implementation details.

user interface frameworks — are contained in — classes; are used to build — source code framework

Basic source code that describes the graphical user interface. Does not include implementation details.

The software development process is iterative, with several rounds of designing, writing, compiling, running, and testing before software is deployed.

text editor — is used to create — source code

source code — is human readable. — is used to create — classes ▸ 12; is used to create — text files

Are human-readable. Usually have the suffixes .java or .jav.

compiler — A software utility that turns human-readable source code into machine-friendly byte code.

byte code — is platform independent. It cannot be read by the processor directly.

binary files — Are computer-readable. Binary files usually have the suffixes .jar, .zcm, .class, .war, or .war.

virtual machine — emulates — byte code — runs on — machine instructions — are processed by — microprocessor

machine instructions — Are processor dependent. They can be read by the processor directly.

fragment or application — may be a — Java applet — runs on — Java-enabled browser — runs on — Java virtual machine

debugger — allows debugging of — objects ▸ 55 — may be changed out of — objects ▸ 55 — helps find — bug — is trapped by ...

Devices — are used by ... support ... services ▸ 26; such as ... hardware ▸ 35

Java technology is especially useful in devices that are connected by a network.

People — may download and install ... JREs ▸ 30; often own ... hardware ▸ 31; some of whom are ... developers ▸ 23 — connects ... devices ▸ 25; supports ... services ▸ 26; supports ... web services ▸ 33

activities — are contained in — services — Including getting information, communicating, collaborating, playing, working, creating, and buying and selling.

services — are contained in — software — Including news services, instant messaging services, e-mail services, communication software, collaboration software, discussion forums, games, the world wide web, productivity software, image-creation software, and marketplace services.

software — is developed with ... platforms

Including servers, web-based solutions, stand-alone applications, and clients.

Programs — such as ... applications ▸ 28; enable ... services ▸ 26; run on and connect ... devices ▸ 25

Java™2 Platform: end-to-end solutions

platforms — provide a stable binary binding to the ... — run on ... — **operating systems**

applications — are contained in — **runnable components** — are configured by — **deployment descriptor** — contains configurations for ... — **containers** — are contained in — **components** — are contained in — **packages or libraries** — are ... class libraries ▸ 10

Software Development Kits (SDKs) — are contained in — **packages or libraries** — may include ... tools ▸ 17; are used in ... development environments ▸ 16; include ... class libraries ▸ 10; are used to create ... programs ▸ 24

Java runtime environments (JREs) — include a ... packages ▸ 11; are used to run ... programs ▸ 24 — run on ... — **operating systems**

A JRE is the software environment in which programs compiled for the Java virtual machine can run. The runtime system comprises everything necessary to run programs written in the Java programming language, including the Java virtual machine, which may be a Java interpreter, and the platform packages. JREs may be developed by Sun or by other companies.

specific devices (hardware)

...for enterprise servers and applications

J2EE™ — depends upon — **enterprise applications** — such as — **web-based applications** — and — **client/server applications** — such as

J2EE™ is used to build and run — Java 2 Platform Enterprise Edition is used to build secure, scalable, distributable server-based applications for large enterprises.

business logic — is described by — **server components** — have interfaces provided by — **client components** — always have — **user interfaces** — provides a — **web browser**

business logic — is focused on — business rules — are configured by — deployment descriptor; are configured by — containers; are managed by — containers; are managed by — Enterprise JavaBeans

can be ... J2SE applets ▸ 38; can be ... J2SE applications ▸ 38; can be ... J2ME applets ▸ 40; can be ... MIDlets ▸ 41

Enterprise JavaBeans™ (EJB) — never directly build — **containers**; never directly build — **EJB container**

Enterprise JavaBeans define independent pieces of business logic or application behavior that can call either servlets, JSPs, or other EJBs.

servlets — may directly build — **web container**; may directly build — **web browser**

Servlets respond to an URL request. Servlets commonly generate HTML, XML, or dynamic images. They can call other servlets, JSPs, or EJBs.

JavaServer Pages™ (JSP™) — may directly build — **web container**; may directly build — **web browser**

JSPs define a page element, which can be HTML, XML, or another markup language. JSPs execute as servlets. They can call other JSPs, servlets, or EJBs.

always directly build — embed programming code in a — **markup language** — may be interpreted by — **web browser**

markup language — is one type of — **Extensible Markup Language (XML)**

Provides a platform-neutral extensible mechanism for organizing and presenting data. Many Java packages and classes utilize XML to store and exchange data.

connector architecture — Provides a mechanism for connecting to legacy information systems.

EJB container — is used to run — **containers**

web container — is used to run — **servlets**; can be created by — **JavaServer Pages™**

Simple Object Access Protocol (SOAP) — Provides a way for applications to communicate. SOAP is typically transmitted via HTTP.

UDDI — Universal Description, Discovery, and Integration provides a way for businesses to discover each other. Can also be used programmatically by applications to locate an application or service.

Sometimes referred to as required or core packages.

J2EE platform — include ... platform packages ▸ 37 — include — **J2EE specific packages**

Examples include: javax.activation, javax.ejb, javax.servlet, javax.transaction, javax.xml

J2EE Software Development Kit — contains — Includes the binary version of the J2EE Reference Implementation, plus the development tools and documentation needed to build a J2EE application.

Java Web Services Developer Pack — is used to build ... web services ▸ 33 — used to build applications — is deployed to — **Java runtime environments**

Provides a framework for developing and hosting J2EE applications that run inside web containers. The pack includes the Java APIs for XML included in the Java XML Pack, the JavaServer Pages Standard Tag Library (JSTL), the Ant build tool, the Java WSDP Registry Server, and the Tomcat Java Servlet and JavaServer Pages container.

Java virtual machine — is instantiated by — **machine** — runs on — **operating systems**

Some JVMs are optimized for servers.

J2EE application server — runs on — **Solaris, Linux, Windows, HP-UX, AIX, FreeBSD** — runs on — **servers**

Hosts EJBs, servlets and JSPs.

Java-enabled web server — runs on — **servers**

Can host EJBs, servlets and JSPs.

Jini™ — Jini network technology provides one infrastructure for delivering services in a network and for creating spontaneous interaction between programs that use those services.

...for desktop servers and applications

J2SE™ — is based on subsets of — **J2ME™**

Java 2 Platform Standard Edition is intended to run on desktops, laptops, and workstations. J2SE also includes the core language packages and classes used by J2EE and J2ME.

applications embedded in a web page — run inside a ... web browser ▸ 34 — are — **desktop applications**

J2SE applets — are built on — **J2SE** — run inside — **web browsers**

Extend the applet class, which provides developers with a basic framework for rapid development of a limited application. This includes a basic user interface framework and event model.

J2SE applications — are built on — **J2SE** — run inside — **desktop applications**

Implement a main method. A class that defines an application must declare a main method, which the Java virtual machine uses as the starting point for the application. The main method gets executed when the application starts.

JavaBeans™ — are built with — **J2SE**

Discrete units of software functionality that conform to the JavaBeans API and are designed to be reusable components. The JavaBeans API makes it particularly easy for Beans to be utilized by visual programming tools. (In an architecture analogy, these would be prebuilt walls, windows, or doors.)

Also referred to as required or core packages, platform packages are the heart of the Java programming language. The packages include myriad classes that define user interface components, a graphics toolkit, net, text and math processing tools, code-level security, objects for memory storage of complex data, and methods for invoking remote applications, to name just a few.

platform packages — are contained in — **J2SE**

Beans — Swing — A component kit that provides a set of graphical user interface elements. Swing can be used to rapidly build applications without writing original code to support common interface elements. Supports a native look and feel on many common operating systems.

Java regular expressions — A powerful set of text matching and manipulation routines supported by many programming languages. Java technology implements Perl-compatible regular expressions.

J2SE optional packages — are — **optional packages** — are — **J2SE**

Optional packages define a set of methods to address specific functionality required for functionally related applications. Applications that require optional packages must be shipped with the package code.

Java 3D™ — A set of classes used to write three-dimensional graphics applets or applications.

JavaHelp™ — A system that enables developers to add interactive online help in applications or web sites.

J2SE Software Development Kit — The J2SE SDK includes the source code, class libraries, development tools, and runtime environment required to build Java applications and applets.

Java Plug-In — may be enabled by a — **Java-enabled browser or viewer** — runs on — **operating system-based Java virtual machine**

Some browsers may be Java-enabled because they include the Java Plug-In. Some users may need to download the Java Plug-In for their browsers. Installing the Java Plug-In will also install the J2SE runtime environment.

Some JVMs are optimized for various types of laptops, workstations, and desktops.

Java virtual machine — runs on — **personal computers**, **workstations**

Windows	SunOS™
Mac OS	AIX
Unix	Alpha
Linux	RS6000
Irix	FreeBSD
Solaris™	NeXT

...for consumer and embedded servants and applications

J2ME™ — is based on subsets of — **J2SE**

Java 2 Platform Micro Edition is intended to run on devices with limited computing power, such as cell phones or PDAs. J2ME has reduced processor and memory requirements. It is based on subsets of J2SE with additional components for market segments such as hand-held devices.

specialized embedded applications — are built on — **J2ME**

PDA applications — are built on — **J2ME**

cell phone applications — are built on — **J2ME**

J2ME applets — are built on — **J2ME** — are built — **J2ME applications**

MIDlets — run in a — **J2ME** — are built on — **J2ME**

Profiles are packages required for a particular vertical market segment or set of related devices. Other profiles include Foundation, Game, Personal Basis Profile, and PDA.

J2ME optional packages — are — **Personal Profile (PP)** — such as — **Profiles** — such as — **MIDP** — depends on — **CDC** — such as — **configurations** — such as — **CLDC** — depends on — **J2ME Wireless Toolkit**

Connected Device Configuration includes specific packages that are subsets of the J2SE equivalent: java.lang, java.util, java.net, java.io, java.text, java.security

A set of packages that define basic services for a range of devices.

There are several implementations of Mobile Information Device Profile, including MIDP for Palm 1.0, 1. MIDP 1.0a for Ninty, and MIDP 2.0.

Connected Limited Device Configuration includes specific packages that are subsets of the J2SE equivalent: java.io, java.lang, java.util, javax.microedition.io

Java virtual machine — runs on — **Palm OS, Windows CE, RTOS, Linux, Symbian OS, proprietary systems** — runs on — **point-of-sale (POS) systems, pagers, cell phones, and personal digital assistants (PDAs)**

A JVM can be optimized for cell phones, PDAs, and other similarly resourced devices.

The J2ME Wireless Toolkit includes the necessary tools, emulators, documentation, and examples to develop and simulate Java applications targeted at cell phones, pagers, PDAs, and other small devices.

Java Card™ — is based on — **J2SE** — run on — **Java Card applets**

applications for use with Smart Cards — Such as ATMs and point-of-sale services. — are built with — **Java Card applets**

Java Card applets — are built with — **Java Card API**

Includes specific packages: java.lang, javacard.framework, javacard.security, javacard.crypto

Java Card virtual machine — runs on — **Windows, Solaris™, proprietary systems** — runs on — **Java Card, subscriber identity module (SIM) phones, and multi-application smart cards**

Includes the necessary tools, emulators, and documentation environment to develop and test applets for the Java Card platform.

Java Card Development Kit

Java TV™ — is built on — **J2SE** — runs on — **interactive television applications** — Such as user-controlled camera views and video on demand. — are built with — **Java TV Xlets**

Java TV Xlets — are built with — **Java TV APIs**

Java TV APIs — are built on — Depend on the Java Media Framework (JMF), which is optional to J2SE but required for Java TV.

Java virtual machine — runs on — **real-time operating systems (RTOS)** — runs on — **televisions**

On handheld and embedded devices, the initial implementation of Java technology addresses the software needs of networked applications running on consumer devices such as set-top boxes and smart phones.

PersonalJava™

Java Embedded Server™ (JES) — is built on — **J2SE** — run on — **embedded applications** — Such as home-services and network-aware automobiles. — are built with — **Java Embedded Server framework**

Specifies packages: http, log, device access, timers, framework

Java virtual machine — runs on — **proprietary systems** — runs on — **network-aware appliances, automobiles, and Telematic networks**

Telematic networks connect computers with telecommunications systems.

A JVM can be optimized for residential gateway servers.

Documentation — is available via http://java.sun.com.

- J2EE documentation — J2EE Platform Specification; J2EE Connector Specification; J2EE API Specification; Java Series; J2EE Tutorials
- J2SE documentation — J2SE Platform Specification; J2SE API Specification; Java Series; J2SE Tutorials
- J2ME documentation — J2ME Platform Specification; Mobile Information Device Profile (MIDP) Specification; Foundation Profile Specification; Connected Limited Device Configuration (CLDC) Specification; Connected Device Configuration (CDC) Specification; PersonalJava Specification; Personal Profile Specification; Personal Basis Profile Specification; J2ME API Specification; Mobile Information Device Profile (MIDP) API Documents; Connected Limited Device Configuration (CLDC) API Documents; Connected Device Configuration (CDC) API Documents; Java Series; J2ME Tutorials
- other documentation — Includes Web Services, Getting Started, Java XML, Internationalization, Sound, and Java 2D.
- tutorials
- Java Language Specification (JLS) — Provides a complete specification for the syntax and semantics of the Java programming language.
- Java BluePrints — Guidelines, patterns, and code for end-to-end applications.

Acknowledgements
Prepared for the Sun Developer Web Services Group by Dubberly Design Office/Devine/Faris/Saddler.

Significant contributions from the following people made this map possible: Josh Bloch, Lisa Friendly, Doug Kramer, Paul Pangaro, Robert Patten, Monica Pawlan, Carla Schroer, Leigh Sheneiki, Alan Sommerer, Ken Tallman, and James Urquhart.

Thanks also to Cindy Bloch, Adam Cohn, Andrey Dikanskiy, Larry Freeman, Dale Green, Joann Gray, Cori Kaylor, Onno Kluyt, Stuart Marks, Shawn Moore, Chuck Mosher, Shaler Ney, Eduardo Pelegri-Llopart, Lori Uzzo, Brad Wetmore, and Hinkmond Wong.

Trademarks
© 2002 Sun Microsystems, Inc. All rights reserved. Sun, Sun Microsystems, the Sun logo, Java, the Java coffee cup logo, J2EE, J2SE, J2ME, Enterprise JavaBeans, EJB, JavaBeans, Java Card, Java Community Process, JCP, Javadoc, Java Embedded Server, JavaHelp, JavaServer Pages, JSP, Java TV, Java 3D, Jini, JVM, PersonalJava, Solaris, and SunOS are registered trademarks of Sun Microsystems, Inc.

Any errors or omissions are the responsibility of the authors. Comments are welcome; contact Audrey Crane at audrey@dubberly.com.

February 14, 2003