

# JDK Glossary of Terms

A work in progress, from my blog at [http://blogs.oracle.com/kto/entry/glossary\\_for\\_jdk\\_builds](http://blogs.oracle.com/kto/entry/glossary_for_jdk_builds).

- 32bit build
  - Builds done on 32bit machines, or creating 32bit native code object files, libraries, and executables. A 32bit JDK build always includes a complete JDK image.
- 6-open
  - Release name for OpenJDK 6.
- 64bit build
  - Builds done on 64bit capable machine and creating 64bit native code object files, libraries, and executables. On Linux 64bit and Windows 64bit, a 64bit build is a complete JDK build, and although a 32bit build could be done on a 64bit Linux or Windows machines, it's not common practice. On Solaris, a 64bit build is just the native code object files, libraries, and executables, which means that a Solaris 64bit build needs to be 'added on' to a Solaris 32bit build. Only on Solaris can you run one JDK and ask for either a 32bit execution (-d32) or a 64bit execution (-d64).
- ARCH\_DATA\_MODEL
  - Makefile variable setting that determines if it is a 32 or 64 bit build. Set to 32 or 64.
- ALT\_ *setting*
  - Makefile variable naming convention used in JDK Makefiles to select alternative paths, locations, or settings.
- ALT\_BINARY\_PLUGS\_PATH
  - Location of OpenJDK binary plugs.
- ALT\_BOOTDIR
  - Location of JDK used to build the JDK, usually the previously released JDK.
- ALT\_COMPILER\_PATH
  - Alternative location for the native compiler to use.
- ALT\_CORBA\_DIST
  - The location of the corba "dist" directory to get the corba files from.
- ALT\_CUPS\_HEADERS\_PATH
  - Location of the CUPS include files.
- ALT\_DEPLOY\_MSSDK
  - Location of Platform SDK installation (Windows only).
- ALT\_DXSDK\_PATH
  - Location of DirectX 9 SDK installation (Windows only).
- ALT\_FREETYPE\_HEADERS\_PATH
  - Location of the freetype native include files.
- ALT\_FREETYPE\_LIB\_PATH
  - Location of the freetype native libraries.
- ALT\_JAXP\_DIST
  - The location of the jaxp "dist" directory to get the jaxp files from.
- ALT\_JAXWS\_DIST
  - The location of the jaxws "dist" directory to get the jaxws files from.
- ALT\_JDK\_IMPORT\_PATH
  - The location of the JDK import image. This is used to fill in the gaps when doing partial builds of the jdk. This should be a recent build of the same version of the JDK you are building.
- ALT\_LANGTOOLS\_DIST
  - The location of the langtools "dist" directory to get the javac files from.
- ALT\_PARALLEL\_COMPILE\_JOBS
  - On Linux and Solaris, the parallel build feature of GNU make is used to build all the object files for a library in parallel, based on this setting. Depending on the number of CPUs you have, this number should be set to something between nCPUS+1 and nCPUS\*2. Each system is different and some experimentation may be required.
- ALT\_PREVIOUS\_JDK\_IMAGE
  - When doing a full control build, one of the last steps is to compare the JDK image you just created with a previous JDK image. This is the image you will compare with. Often this is the previous JDK release, and could be the same as ALT\_BOOTDIR.
- ALT\_SLASH\_JAVA
  - Many of the default paths for the various ALT\_\* settings are rooted at the path "/java/" or "J:/" on Windows. This variable allows you to change that root in case you have created an exact copy of the "/java/" area in a local disk area or faster access location. Unfortunately, this layout isn't completely documented and may need to be deduced from the output of 'gnumake sanity' output.
- BOOTDIR
  - See ALT\_BOOTDIR.
- BUG
  - See Change Request.
- BugID
  - See Change Request.
- CCC
  - JDK compatibility process for developers. TBD for OpenJDK.
- CR
  - See Change Request.
- Change Request
  - A Change Request or CR (also referred to as a bug, rfe, or bugid) is an entry in the Sun bug tracking system. These can be seen partially from <http://bugs.sun.com>. The OpenJDK bug tracking system is TBD, but for now CR's will have to do. A CR number is a 7 digit number, and one of these is required for every change made to the JDK sources. If you see the 7 character pattern "NoBugID" used, it means that no official CR was created for this change, you should rarely if ever see them.
- Core libraries
  - A subset of the JDK is considered to be the Core Libraries.
- DSCM
  - Distributed Source Code Management (SCM) system. Some examples are TeamWare, Mercurial, and git. Users can 'clone' or create child repositories or workspaces that are complete copies allowing for distributed development that can optionally be merged back into the primary branch of development or not. CVS and SubVersion are not distributed, there is only one instance of a repository.

- GTEE
  - An internal Sun testing grid system used to test JDK promotions and releases.
- HotSpot
  - The JDK VM or JVM. There are various implementations of the HotSpot VM: client, server, tiered, and kernel.
- Hotspot Express
  - A new release model which somewhat disconnects the Hotspot VM releases from the JDK releases. Depending on the JDK release or the update release, the Hotspot team will choose which version of Hotspot will be integrated into that JDK. This creates a separate version and build number for the Hotspot VM inside the JDK, seen when you run 'java -version' saying something like "build 11.0-b15" or Hotspot version 11 build 15.
- JAVA\_HOME
  - An environment variable used to trigger the 'java' found in your PATH to use a different JDK image. Unfortunately, not all 'java' startup scripts obey this env variable. It's also used by many java tool startup scripts to determine what 'java' to run, bypassing the 'java' found in the PATH setting. Setting this variable during a JDK build is a bad idea, don't do it.
- JDK\_IMPORT\_PATH
  - See ALT\_JDK\_IMPORT\_PATH.
- JPRT
  - An internal Sun build system that was created from the hotspot specific PRT build system.
- MASTER
  - The name given to the primary or master copy of the sources for a release. It is the MASTER sources that JDK promotion builds are built from.
- NoBugID
  - See Change Request.
- PIT
  - A formal testing process that integrators are required to go thorough prior to integrating their source changes into the MASTER area.
- Platform SDK
  - Also spelled "PlatformSDK" or just "PSDK". This is a Platforms specific SDK kit that must be installed on Windows systems in order to build the JDK. With Windows 32bit, the Visual Studio 2003 compiler installation may include at least a partial PlatformSDK image, but the full Windows Platform SDK may also need to be installed on 32bit systems. On Windows 64bit systems, the PlatformSDK is critical because the native compiler for 64bit will come from this SDK. The specific version of the PlatformSDK you install is critical, and depends on the version of the JDK you are building.
- PRT
  - See JPRT.
- PSDK
  - See Platform SDK.
- RFE
  - See Change Request.
- SCCS
  - Source Code Control System. Unix per-file source code change system. Used by TeamWare to manage the changes to individual files.
- SCCS keyword
  - A pattern of a '%' uppercase letter and another '%', e.g. %W%, that when used in a source file controlled by SCCS would get expanded on a 'sccs get' operation and not expanded during the edit of a source file. Lots of different types of keywords were available, e.g. version numbers of files, dates of changes, etc.
- SCCS revision
  - The commit of a change to an SCCS file creates an SCCS rev for that file.
- SDK
  - Software Development Kit. A horribly overused term which means a big bag of stuff that is usually associated with helping you develop some kind of software that uses a particular API or platform.
- SOS8
  - Version 8 of the Sun Studio native compilers and tools. This version was used in the jdk5 release.
- SOS9
  - Version 9 of the Sun Studio native compilers and tools. This version was never used for any jdk releases.
- SS10
  - Version 10 of the Sun Studio native compilers and tools. This version was used for the Solaris X64 bits of jdk5, and for the initial builds of jdk6 before it was released.
- SS11
  - Version 11 of the Sun Studio native compilers and tools. This version was used for the jdk6 release, and what is being used now for the jdk7 builds.
- SS12
  - Version 12 of the Sun Studio native compilers and tools. Most recent release, and will probably be one used to build jdk7. This release also has Linux X86 compilers, use of the Linux release is being investigated.
- VC6
  - The old Microsoft C/C++ compilers and tools used to build jdk5 on Windows 2000 32bit.
- VS2003
  - Visual Studio 2003 .NET. Used to build jdk6 on Windows 2000 32bit.
- VS2005
  - Visual Studio 2005 .NET. Some kind of preliminary VS2005 64bit version was part of the PlatformSDK used to build jdk6 and jdk7 Windows 2003 64bit builds.
- ant
  - Build tool used by some parts of the jdk7 sources.
- approval
  - Term used to indicate either approval of a webrev by a fellow engineer, or higher level approval by a group that indicates the changes have been approved for putback or integration.
- binary plugs
  - The missing parts of the JDK whose source is not available in the OpenJDK. These are quickly being replaced with open source versions, so the hope is that some day we would talk about these in the past tense, e.g. "Remember when we had those painful binary plugs?". :^)
- bringover
  - TeamWare action of pulling changes from one workspace to another. Conflicts are allowed and would have to be resolved.
- build flavor

- Usually "product", "debug", or "fastdebug", and reflects how the source code was compiled. The product build flavor is usually optimized code with no assert type checking left in the product and little debugging information. The debug build is usually not optimized at all, has all assert checking in place and complete debug information. The fastdebug build is a debug build but with as much optimization of the code as feasible.
- ccc
  - See CCC.
- changeset
  - Mercurial term used to describe a complete change. Can include file changes, file deletes, file renames, and file additions.
- code review
  - See webrev and review.
- commit
  - Mercurial action that actually creates a formal changeset.
- conflict
  - When separate changes are made to the same file and they have to be resolved or merged.
- control
  - The name of the jdk workspace that controls the complete jdk build.
- control build
  - Using the control workspace to build the jdk.
- corba
  - The jdk workspace that contains all the corba contributions to jdk7.
- cygwin
  - Windows open source software than can be used to build the jdk.
- deploy
  - The jdk workspace that contains the plugins and java webstart.
- extension
  - Mercurial term for adding functionality, e.g. the "forest extension".
- findbugs
  - Software that can be used to find common bug patterns in classfiles and Java sources.
- gatekeeper
  - The person responsible for pushing changes from a shared team area to the MASTER repositories. See [http://blogs.oracle.com/wetmore/entry/you\\_re\\_a\\_gatekeeper\\_uh](http://blogs.oracle.com/wetmore/entry/you_re_a_gatekeeper_uh) for more about what it means to be a gatekeeper.
- gcc
  - The GNU C compiler, sometimes also refers to the GNU C++ compiler g++.
- header files
  - See include files.
- hg
  - The Mercurial command name.
- hgrc
  - Mercurial resource file. Users have a ~/.hgrc file, repositories have a .hg/hgrc file, and the installation of Mercurial has a etc/mercurial/hgrc file.
- hooks
  - Mercurial feature that allows repository actions to be checked or monitored.
- images
  - After building a jdk, the Makefile target "images" is used to create the actual jdk and jre install images.
- include files
  - Native code files that contain global type, variable and/or function declarations and prototypes.
- install
  - The name of a closed workspace that creates the official install bundles, but also a general term for the act of installing something. DUH.
- install image
  - The actual jdk disk image that represents what the end user would see after an install. Typically contains the following directories: bin, lib, include, man, and demos.
- integration
  - The act of doing the putback or push of the changes into a shared team or MASTER source tree area.
- javah
  - The javah tool is run over class files to create header or include files that can be used with the native code to implement the "native" java methods.
- j2se
  - Java 2 Standard Edition. Old name, don't use it anymore. Use jdk. Never let marketing control your version numbers or product name. ; ^) This is also the name of the primary jdk workspace that built most of the jdk classes and jni support libraries (minus hotspot and the plugins).
- jaxp
  - The jdk workspace that contains the JAXP team contribution to jdk7. In jdk6 and jdk5, these sources were part of the j2se workspace.
- jaxws
  - The jdk workspace that contains the JAX-WS team contribution to jdk7. Contains JAXB and JAF classes too.
- jdk
  - Java Development Kit or Jim's Designer Kitchen.
- jdk5
  - The 1.5 release of the jdk, also jdk 1.5, jdk 1.5.0, or just jdk 5.
- jdk5 update
  - Update release of jdk5. DUH.
- jdk6
  - The 6 release of the jdk, also jdk 1.6.0, or just jdk 6.
- jdk6 update
  - Update release of jdk6. DUH.
- jdk7
  - The current jdk release.
- jplan
  - Sun internal tool used for jdk planning.
- jtreg

- Tool available on OpenJDK site, used to run the regression tests in the OpenJDK sources. A command script over the JavaTest framework.
- langtools
  - The jdk workspace that contains the javac team contribution to jdk7. In jdk6 and jdk5, these sources were part of the j2se workspace.
- mercurial
  - Mercurial DSCM. DUH.
- merge
  - Mercurial term used to merge changesets. With Mercurial there needs to be one changeset that represents the state of the entire repository, so when changesets mix, a merge changeset needs to be created so we have one changeset again.
- mks
  - Unix tools for Windows than can be used instead of cygwin for building the jdk.
- motif
  - A former jdk workspace that is no longer needed in jdk7.
- native code
  - C, C++, or assembly code that creates native object files, libraries, or executables. Header files or include files are also considered native code.
- openjdk
  - Original open source of the jdk.
- openjdk6
  - The 6-open release of the OpenJDK.
- openjdk7
  - The original OpenJDK release that will be used to create jdk7.
- partial build
  - Any build of a jdk that doesn't include the complete set of jdk sources. A partial build will typically rely on a JDK\_IMPORT\_PATH setting to import the non-built parts of the jdk.
- platform build
  - Building on a particular platform. The current platforms that the jdk7 is built on regularly are: Solaris 10 X86, Solaris 10 X64, Solaris 10 SPARC, Solaris 10 SPARCV9, Windows 2000 X86, Windows 2003 X64, Linux RedHat 2.1AS X86, and Linux SuSE SLES-8.1 X84. But jdk7 is buildable on many more platforms than these.
- pubs
  - Source of jdk documents. Currently not "open".
- pull
  - Mercurial command that pulls changesets from one Mercurial repository to another repository.
- push
  - Mercurial command that pushes changesets from one Mercurial repository to another repository.
- putback
  - TeamWare command that pushes changed SCCS files from one TeamWare workspace to another workspace.
- regression tests
  - In the jdk, tests that are in the source tree test directory. Policy dictates that any bug fix should have an associated regression test to verify that the fix works.
- repository
  - The term Mercurial uses for a managed source tree.
- resolve
  - TeamWare command used to resolve conflicts in file changes between different SCCS revisions.
- review
  - The action of doing a code review, usually using a webrev.
- SCM
  - Source Code Management system.
- server
  - Horribly overused term that in this case indicates a machine that holds the files of a workspace or repository.
- slot
  - Integrations into MASTER are treated special. Integrators into the MASTER area are given a time "slot", usually a AM or PM time slot where they have putback or push rights to the MASTER area reserved. During this time the integrator would be expected to sync, build, and test prior to integrating his changes into the MASTER sources.
- source bundle
  - A zip or tarball of a source tree.
- source tree
  - The sources that are managed by a SCM, minus any SCM support files or directories.
- sponsors
  - The name for a closed jdk workspace.
- ssh
  - Secure shell. A more secure and better way to access machines on a network. Replaces things like telnet, rsh and rlogin.
- ssh key
  - The basic way ssh works is with keys. Run ssh-keygen to create your ssh key.
- SunJDK
  - Sun's JDK, available from <http://java.sun.com>.
- sync
  - The action of pulling the latest changes from a MASTER or team integration area into your own changes and merging it all together.
- teamware
  - TeamWare is a distributed source code management system. A Sun product.
- webrev
  - A script or tool that creates browsable pages for a change to a managed source tree. Also used to describe these browsable pages, e.g. using webrev to create a webrev. Used for all code reviews, and all code changes must be reviewed. With Mercurial, browsing a changeset is very similar to browsing a webrev.
- workspace
  - TeamWare term for s managed source tree.