JDK 7 新功能和增强功能

Java SE 7 Features and Enhancements

Java Platform, Standard Edition 7 is a major feature release. This document includes information on features and enhancements in Java SE 7 and in JDK 7, Oracle's implementation of Java SE 7.

Contents

技术变革的亮点

Highlights of Technology Changes in Java SE 7

The following list contains links to the the enhancements pages in the Java SE 7 guides documentation. Choose a technology for further information.

- Swing
- IO and New IO
- Networking
- Security
- Concurrency Utilities
- Rich Internet Applications (RIA)/Deployment
  - Requesting and Customizing Applet Decoration in Dragg able Applets
  - Embedding JNLP File in Applet Tag
  - Deploying without Codebase
  - Handling Applet Initialization Status with Event Handlers
- Java 2D
- Java XML - JAXP, JAXB, and JAX-WS
- Internationalization
- java.lang Package 多线程自定义类加载器
  - Multithreaded Custom Class Loaders in Java SE 7
- Java Programming Language Java 编程语言
  - Binary Literals
  - Strings in switch Statements
  - The try-with-resources Statement 资源自动关闭
  - Catching Multiple Exception Types and Rethrowing Exceptions with Improved Type Checking 捕获多个异常并重新抛出异常
  - Underscores in Numeric Literals 数字字面常量中的下划线
  - Type Inference for Generic Instance Creation 泛型实例创建的类型推导
  - Improved Compiler Warnings and Errors When Using Non-Reifiable Formal Parameters with Varargs Methods

Important RFEs Addressed in Java SE 7

This list includes some of the notable RFEs that relate to Java SE 7. Changes to Java SE 7 include changes to the Java language, the definition of the Java Virtual Machine (JVM), or the Java SE API Specification.
Changes to JDK 7, Oracle's implementation of Java SE 7, are presented in the Important RFEs Addressed for JDK 7 section.

Area: HotSpot
Synopsis: The JVM/TI version number was changed from 1.1 to 1.2 in order to add the `GetLocalInstance` method.
RFE: 7003782, 7004582

Area: Security
Synopsis: Security algorithm requirements have been defined for Java SE 7 that provide a list of algorithms that all implementations of Java SE 7 must support. The class summary of applicable classes (ex: `java.security.Signature`) has been updated to include the implementation requirements. Also, all of the requirements are listed in the Implementation Requirements section of the Standard Algorithms document.
RFE: 5001004

Area: API: JSSE
Synopsis: In previous releases, except for the default algorithm, there was no standard algorithm name for `KeyManagerFactory`. In the Java SE 7 release, "PKIX" is exported as the standard algorithm name for `KeyManagerFactory`.
The "PKIX" `KeyManagerFactory` algorithm is defined as:
A factory for X509ExtendedKeyManagers that manages X.509 certificate-based key pairs for local side authentication according to the rules defined by the IETF PKIX working group in RFC 3280 or its successor. The KeyManagerFactory must support initialization using the class `javax.net.ssl.KeyStoreBuilderParameters`.
RFE: 7022855

Area: API: JSSE
Synopsis: Support for TLS 1.2 has been added to the SunJSSE provider.
RFE: 6916074

Area: API: Language
Synopsis: Class loading can be prone to deadlocks if custom class loaders do not adhere to an acyclic class loader delegation model. New APIs have been added to the `java.lang.ClassLoader` class to support parallel loading of classes and finer-grained locking mechanism for class loading operations. Custom class loaders which would like to leverage this functionality must refer to the Class Loader API Modifications for Deadlock Fix documentation for the suggested model and requirements and be implemented accordingly.

Area: API: AWT

Synopsis: The `java.awt.Cursor` class contained a "protected static field" called "predefined." This field has been removed.

Area: API: AWT
Synopsis: Color alpha values are now preserved by `java.awt.Color.darker()` and `java.awt.Color.lighter()` methods.
RFE: 6783910

Important RFEs Addressed in JDK 7

This list includes some of the notable RFEs that relate to JDK 7, Oracle's implementation of Java SE 7. Changes to JDK 7 includes changes to `javac`, to HotSpot (and related tools), and to the implementation of the Java SE 7 API. Changes to Java SE 7 are presented in the Important RFEs addressed for Java SE 7 section.

Area: Scripting
Synopsis: The JDK 7 release is co-bundled with the Mozilla Rhino JavaScript engine based on version 1.7R3 pre-release sources with Oracle modifications. You can download the Oracle modified Rhino sources at http://jdk7.java.net/rhino/.

Area: Tools
Synopsis: A succinct and concise error message is printed, as appropriate, when an error occurs during the Java launcher initialization. The full stack trace will be printed when applicable, or when the error has trickled up from the ClassLoader or Reflection APIs.
RFE: 6968053

Area: Tools
Synopsis: Previously, the pack200 tool defaulted to using java6 packfiles in the absence of any java6 (or newer) classfiles in a segment. Starting with JDK 7, the pack200 tool defaults to using java5 packfiles for maximum compatibility.
RFE: 6712743

Area: Tools
Synopsis: On Solaris only, the JRE and JDK distributions had the following symbolic link (symlink): `jre/lib/libjvm.so->client/libjvm.so`. This was provided as a transition aid from Solaris Production JRE to the Java Reference version (more precisely ExactVM to HotSpot). This symlink will cease to exist and is no longer supported.
Applications (usually legacy Solaris applications with a custom launcher) that have assumed the existence of this symlink, must relink with the desired library (implementing server or client), or must dynamically link with the desired VM. They can also create that symlink manually or copy the desired library implementing the VM, though this is not a recommended practice and may cause undesired side effects.
RFE: 6899834

Area: Tools
Synopsis: The HtmlConverter tool is no longer part of the JDK 7 distribution (starting with b128).
RFE: 4523289

Area: Tools

Synopsis: The default values for the cryptographic algorithms used by the `keytool` and `jarsigner` utilities have been updated to be stronger. See the `keytool` and `jarsigner` tool docs for more information.
RFE: 6561126

Area: Tools
Synopsis: The `keytool` and `jarsigner` tools now support the ECC algorithm in keypair generation and jar signing.
RFE: 6870812

Area: Tools
Synopsis: A new `keytool` command (`-gencert`) is provided to generate a non self-signed certificate. A new option (`-ext`) is provided to generate several X.509 v3 extensions for the `-gencert`, `-genkeypair`, and `-certreq` commands. There are also other new commands and options for the `keytool` and `jarsigner` tools. See the `keytool` and `jarsigner` tool docs for more information.
RFE: 6780416, 6802846, 6324292, 6890872

Area: Tools
Synopsis: In JDK 7, the `javadoc` tool has been updated to use style markup classes that can control the styles of the generated pages. With the `-stylesheetfile` option, you can provide an alternate stylesheet file to be used. Without this option, the `javadoc` tool automatically creates a stylesheet file called "stylesheet.css". You can override this default with another file name, for example:

```
    C:> javadoc -stylesheetfile C:\user\exampleStylesheet.css com.examplePackage
```

The generated pages have style markups for various sections. Through the stylesheet file you can define styles for these sections. The default stylesheet file is sorted according to each style in that section. You can customize the styles for:

- Page navigation
- Headers and Footers
- Content
- Overall document

RFE: 7052425, 6851834

Area: HotSpot
Synopsis: Classfiles with version number 51 are exclusively verified using the type-checking verifier, and thus the methods must have `StackMapTable` attributes when appropriate. For classfiles with version 50, the HotSpot JVM would (and continues to) failover to the type-inferencing verifier if the stackmaps in the file were missing or incorrect. This failover behavior does not occur for classfiles with version 51 (the default version for JDK7).
Any tool that modifies bytecode in a version 51 classfile must be sure to update the stackmap information to be consistent with the bytecode in order to pass verification.
RFE: 6693236

Area: HotSpot
Synopsis: In JDK 7, interned strings are no longer allocated in the permanent generation of the Java heap, but are instead allocated in the main part of the Java heap (known as the young and old generations), along with the other objects created by the application. This change will result in more data residing in the main Java heap, and less data in the permanent generation, and thus may require heap sizes to be adjusted. Most applications will see only relatively small differences in heap usage due to this change, but larger applications that load many classes or make heavy use of the `String.intern()` method will see more significant differences.
RFE: 6962931

Area: HotSpot
Synopsis: The default out-of-the-box heap size and shape parameters for the concurrent mark sweep collector (CMS) have been modified. The new settings take advantage of faster platforms that have been introduced since

JDK 6 was released. Like the other collectors in HotSpot, CMS will now use the available physical memory on the platform to size its heap, while attempting to shape that heap to keep pause times associated with minor collections "reasonable". The specific shape of the heap may be platform-dependent in other ways as well. Users can override all or some of these default settings by explicitly sizing or shaping the heap, AKA "heap tuning", to suit their specific needs.
For more information on the default settings for this and other garbage collectors, please see the Heap Tuning Guide for JDK 6.
RFE: 6896099

Area: HotSpot
Synopsis: The JDK 6 release included performance enhancements for CMS. To provide a mechanism to continue using the behavior provided by JDK 5, the `CMSUseOldDefaults` flag was provided. This flag restored a number of settings to a default state. During the last few years, this flag has received minimal use and most customers prefer the improved CMS performance. The `CMSUseOldDefaults` flag is now removed.
RFE: 7027529

Area: HotSpot
Synopsis: The GarbageFirst (G1) garbage collector is experimental in this release. Some command line tools, such as `jstack` or `jmap`, may not work properly when using the G1 collector.
RFE: 6966967

Area: HotSpot
Synopsis: In previous releases of the JDK on Solaris only, the `Thread.interrupt()` method would interrupt some blocking I/O operations resulting in `InterruptedIOException` thrown by the target thread and leaving socket or file streams in an inconsistent state. This so called "legacy interruptible I/O support" has been disabled in JDK7. Applications that previously relied on this Solaris specific behavior can re-enable this support by running with the following option on the command line: `-XX:+UseVMInterruptibleIO`. A future release of the JDK may remove the legacy interruptible I/O support completely.
RFE: 6554406

Area: JSSE
Synopsis: The TLS renegotiation fix has been implemented. Please see TLS / SSLv3 Renegotiation Vulnerability Explained, Understanding the TLS Renegotiation Attack and Authentication Gap in TLS Renegotiation for more information.

Area: JSSE
Synopsis: Support for the Server Name Indication (SNI) extension has been added to the JSSE client in the SunJSSE provider.
RFE: 6985179

Area: SASL
Synopsis: NTLM is now supported as a SASL mechanism on both the client and server side. Only the authentication layer is implemented, and there is no privacy or integration protection in communication.
RFE: 6911951

Area: Installer
Synopsis: As of the Solaris 11 Express release, the IPS packaging system is the new default mechanism for installing, upgrading, and removing software packages. The JDK 7 release supports the IPS system. IPS packaging introduces change to all users of Solaris. Every package will likely have a name change and process change. Automated installers and install verification applications written by users or third parties will require updating.
RFE: 6979976

Area: Installer
Synopsis: As of JDK 7, we are no longer distributing .sh or .bin packages. Refer to the Troubleshooting guide for

more information.
RFE: 6980760

Area: Installer
Synopsis: On SUSE Linux SLES10 x86, the 32-bit JRE/JDK can not be installed. Refer to the Troubleshooting guide for more information.
RFE: 7006936

Area: Installer
Synopsis: The JRE can not be installed on Solaris/Linux if the previous version of the release is already installed. The workaround is to specify the `-F` option.
RFE: 6978307

Area: Installer
Synopsis: Under OLS6 x64, the 32-bit version of the JDK and JRE can not be installed; the `java.exe` version fails.
RFE: 7005996

Area: Deployment
Synopsis: Previously, the pack200 tool segmented its output by default. As of this release, the pack200 tool will create one large segment per jar file. Therefore, if deployers have jar files larger than the virtual memory available on the end-user's systems, it is recommended that either the input jar file be split, or suitably segmented by using the command-line flag `"--segment-limit=nnnnn"` or the equivalent property "`SEGMENT_LIMIT`".
RFE: 6575373

Area: Deployment
Synopsis: On Windows XP machines, the default cache directory is now under `$USER\Local Settings\Application Data\Sun\Java\Deployment\cache`. If needed, you may customize it to point to a network share folder for a single application cache across domain machines.
RFE: 7012538

Area: Deployment
Synopsis: The optional applet parameter, `jnlp_embedded`, provides the option to cache JNLP content on the HTML page and shortens applet launch time by skipping network access. The `jnlp_embedded` parameter has as its value the base64 encoded content of the applet JNLP file. For example:
```
<applet  width="710" height="540" >
           <param name="jnlp_href" value="launch.jnlp"/>
           <param name="jnlp_embedded"
                value="PD94bWwgdmVyc2lvbj0iMS4wIiB . . . dC1kZXNjPg0KPC9qbmxwPg0
K"/>
           <param name="draggable" value="true"/>
       </applet>
```
When present, the `jnlp_embedded` parameter value replaces the content of the JNLP value pointed to by the `jn.p_href` parameter. The value of `jnlp_ref` is then optional and only used as a backup when the content of `jnlp_embedded` is invalid. There are a few restrictions on the embedded JNLP content:

- The attribute `href` from element `jnlp` should be relative.
- The attribute `codebase` from element `jnlp` should be empty (which means the codebase value will be derived from the document base URL).

RFE: 6990877

Area: Deployment
Synopsis: As of JDK 7, clearing the cache works as follows:

- Only non-installed resources are removed from the cache when invoking `javaws -XClearCache`.
- Both installed and non-installed resources are removed by invoking `javaws -uninstall`.

- The "Delete Temporary Files" dialog, in JCP, previously had two checkboxes, both checked by default: "Applications and Applets", and "Trace and Log files". As of JDK 7, there will be three checkboxes, with the first two checked by default: "Trace and Log Files", "Cached Applications and Applets", and "Installed Applications and Applets".
- Only installed applications will have an entry in the Add/Remove panel.

RFE: 6873615

Area: JNLP Files
Synopsis: In order to support debugging of Java Web Start applications, the `-XX:HeapDumpOnOutOfMemoryError` flag is now supported in JNLP files for trusted applications.
RFE: 6664424

Area: JNLP Files
Synopsis: Previous versions of Java Web Start did not correctly implement section 6.0.10 of the JNLP Specification. With this fix, fine-grained values for the "os" attribute, such as `os="Windows\ XP"`, `os="Windows\ Vista"`, and `os="Windows\ 7"`, will work as expected. Values such as `os-"Win"`, and `os="Windows"` will continue to match all Windows platforms. As of this release, `os="Windows\ Vista Windows\ 7"` will only match Vista or Windows 7 and not Windows XP.
RFE: 7014170

Area: Plugin
Synopsis: A message stating that the first generation of the Java plugin is deprecated is now printed to the log file and to the Java console whenever plugin 1 is used.
RFE: 7027792

Area: Plugin
Synopsis: On Windows, the tray icon is now disabled by default. To enable the tray icon, use the Windows Start menu.
RFE: 6694710

Area: Plugin
Synopsis: The 64-bit toolkit is now supported on 64-bit Windows platforms.
RFE: 6492139

Area: Plugin
Synopsis: Previously, the persistence API provided only temporary persistence for applets. When the VM exited, the data was lost. Data can be persisted via any of the standard Java mechanisms, through the use of LiveConnect to store data in the page DOM, or using one of the JNLP services.
RFE: 6992419

Area: Plugin
Synopsis: Google Chrome is now supported by the DT Plugin.
RFE: 6907245

Area: Security
Synopsis: The implementation of PKIX has been enhanced to include an option to reject certificates if the corresponding key is not strong enough, for example MD2 hash functions or any RSA key with key size less than 1024.
RFE: 6792180

Area: API: JSSE
Synopsis: Support for TLS 1.1 has been added to the SunJSSE provider, and the SSLv2Hello "pseudo protocol" is

no longer active by default in the SunJSSE provider.
RFE: 4873188

Area: API: Language
Synopsis: Previously, the two exception types in the `javax.lang.model.type` package, `MirroredTypeException` and `MirroredTypesException`, were unrelated. In the `javac` implementation, `MirroredTypeException` was thrown where `MirroredTypesException` should have been thrown. In part to address this problem, `MirroredTypeException` was made a subclass of `MirroredTypesException`. This change is binary compatible and generally preserves the behavior of existing annotation processors. However, it is possible this change may cause source incompatibilities for client programs; in those cases, changing the order of catch clauses should allow the programs to compile again.

Area: API: Language
Synopsis: To model the language changes in this release, several updates were made to `javax.lang.model.*` including adding a method to the `javax.lang.model.type.TypeVisitor` interface. Such an addition is source incompatible with libraries that have directly implemented the `TypeVisitor` interface. However, such additions were foreseen as part of this API's evolution and libraries were explicitly cautioned to extend one of the utility visitors instead of directly implementing such an interface.
RFE: 6933147

Area: API: Utilities
Synopsis: Due to an error in `java.util.TreeMap`, it was previously possible to insert invalid null elements and elements not implementing `Comparable` into empty `TreeMaps` and `TreeSets`. Only a single invalid element could be inserted into the empty `TreeMaps` or `TreeSets`; additional elements would cause the expected `NullPointerException` or `ClassCastException`. Most other operations upon the collection would also fail. As of JDK 7, inserting an invalid null element or an element not implementing `Comparable` into an empty `TreeMap` or `TreeSet` throws a `NullPointerException`.
RFE: 5045147

Area: API: NIO
Synopsis: Prior to the JDK 7 release, direct buffers allocated using `java.nio.ByteBuffer.allocateDirect(int)` were aligned on a page boundary. In JDK 7, the implementation has changed so that direct buffers are no longer page aligned. This should reduce the memory requirements of applications that create lots of small buffers. Applications that previously relied on the undocumented alignment can revert to previous behavior if they are run with the command line option: `-XX:+PageAlignDirectMemory`.
RFE: 4837564

Area: API: AWT
Synopsis: Limited mixing of heavyweight (e.g. AWT) and lightweight (e.g. Swing) components is now officially supported. Details are available in the API specification (see the `java.awt.Component` class specification and its "See Also" link), and the Mixing Heavyweight and Lightweight Components article. If an application uses custom code to address the mixing issue and experiences problems with the built-in support for mixing, the feature can be turned off by specifying the `-Dsun.awt.disableMixing=true` system property.
RFE: 4811096

Area: API: AWT
Synopsis: The mechanism used to detect whether the window manager supports a system tray (and therefore the `TrayIcon` API) has been changed. Instead of looking at the current window manager, the System Tray Protocol Specification is used.
RFE: 6438179

Area: API: AWT
Synopsis: In the JDK 7 release there are two AWT toolkits that are supported: `WToolkit` on Windows and `XToolkit` on Linux/Solaris. The `MToolkit` implementation on Linux/Solaris is no longer supported.

Area: API: AWT
Synopsis: With this fix, all `java.awt.Window` objects (not frames or dialogs) are regular top-level windows on X11. In the JDK 6 release, they were `OverrideRedirect` windows.
RFE: 6380835

Area: API: AWT
Synopsis: Windows that enable the `PERPIXEL_TRANSLUCENT` kind of transparency by means of setting a non-opaque background color may not display heavyweight (e.g. AWT) child components correctly. AWT supports lightweight (e.g. Swing) components in per-pixel translucent windows only. More information on the difference between the two kinds of components can be found in the `Component` class specification and in the Mixing Heavyweight and Lightweight Components article.
RFE: 6802853

Area: API: DnD
Synopsis: Because of some security enhancements, some operations for untrusted applications throw or print exceptions. For instance, if during a drag and drop operation the mouse pointer is dragged over an untrusted Java application, the application can retrieve the transferred data only in the drop callback method. Another example is the the clipboard implementation. In an untrusted application, the user cannot access the system clipboard without the `java.awt.AWTPermission.accessClipboard` permission.

Area: JGSS
Synopsis: Java now reads a keytab file whenever that keytab file changes. The file can be empty or nonexistent when the application that uses the file is started.

Area: JGSS
Synopsis: A default configuration file is now provided for JGSS with default `krb5.conf` settings for Windows and *nix systems. This makes deploying a JGSS/krb5 program very easy, especially for deploying Java applets.
RFE: 6483218, 6785456, 6552334

Area: JCE
Synopsis: SunPKCS11 provider now supports Raw RSA encryption, for example, `Cipher.getInstance("RSA/ECB/NoPadding")` calls, when the underlying PKCS11 library supports CKM_RSA_X_509 mechanism. In addition, SunPKCS11 provider recognizes "RSA" as an alias for the "RSA/ECB /PKCS1Padding" transformation when requesting a `Cipher` object.
RFE: 6994008

Area: JCE
Synopsis: SunPKCS11 provider now supports ECB, CBC modes with PKCS5Padding for certain block ciphers. To be more specific, it supports the following transformations for `Cipher.getInstance(...)` calls when the corresponding PKCS11 mechanism is supported by the underlying PKCS11 library:

```
DES, DESede, AES, and Blowfish with CBC mode and PKCS5Padding
DES, DESede, AES with ECB mode and PKCS5Padding
DES, DESede, AES with ECB mode and NoPadding
```

RFE: 4898461

Area: JCE
Synopsis: The SunPKCS11 provider now supports AES encryption with Counter mode (CTR) (for example, `Cipher.getInstance("AES/CTR/NoPadding")` calls) when the underlying PKCS11 library supports CKM_AES_CTR mechanism.
RFE: 6604496

Area: JCE
Synopsis: Given that the relevant Solaris bug (i.e. 6306708 "CKM_SSL3_KEY_AND_MAC_DERIVE returns incorrect cipher keys for exportable ciphersuites") has been addressed in the Solaris 10 update 5 release, the

SunPKCS11 provider no longer disables the following two mechanisms by default:

- `CKM_SSL3_KEY_AND_MAC_DERIVE`
- `CKM_TLS_KEY_AND_MAC_DERIV`

RFE: 7036252

Area: Java DB
Synopsis: Update JDK7 with Java DB 10.8.1.2
RFE: 7042197

Known Issues

Area: Tools
Synopsis: The compiler erroneously accepts array creation expressions featuring the diamond operator. For example:
```
Object[] o = new ArrayList<>[42];
```
This program is ill-formed (according to the JLS) and should therefore be rejected.
RFE: 7057297

Area: Tools
Synopsis: On Solaris and Linux, usage of the `LD_LIBRARY_PATH` environment variable has been purged from the Java launcher. This fix is a seamless change for most Solaris and Linux users, but Java applications invoking JDK 7 from a legacy JDK must be careful to clean up the `LD_LIBRARY_PATH` environment variable before executing JDK 7, or simply use JDK 7 to invoke JDK 7. For more information, see the blog entry, Purging LD_LIBRARY_PATH.
RFE: 6367077

Area: Tools
Synopsis: On Solaris and Linux, with the purging of the `LD_LIBRARY_PATH` environment variable, the JDK 7 launcher is vulnerable to the `LD_LIBRARY_PATH` value that might be picked up when a JDK6 executable invokes a JDK 7 executable and the settings in the parent executable are available to the child executable. In order to mitigate this scenario, the Java launcher will protect itself by setting `LD_LIBRARY_PATH` and variants on Solaris, if the environment variable `LD_LIBRARY_PATH` contains a path to a legacy runtime.
RFE: 7029048

Area: Tools
Synopsis: The `javac` compiler may throw an exception if a class uses an annotation that requires a value, and the value is the first seen reference to a constant value in another class, and which is itself annotated. To work around the problem, add a static import for the constant value, so that its use as an annotation value is not the first seen reference to that value.
RFE: 7043371

Area: Tools
Synopsis: The following program no longer compiles:
```
import java.util.*;

interface A { List<Number>  getList(); }
interface B { List getList(); }
interface AB extends A, B {}
```

```
class Test {
    void test(AB ab) {
        Number n = ab.getList().get(1); //error here
    }
}
```

This is due to the fact the `ab.getList()` used to be resolved as `A.getList()`, while now is resolved as `B.getList()`. Meanwhile, the following workaround can be used:

```
      Number n = ((A)ab).getList().get(1); //works
```

RFE: 7062745

Area: HotSpot
Synopsis: This loop optimizer problem occurs only rarely and has been present since at least JDK 7 build 39 and JDK 6 Update 14. It occurs when an on-stack replacement (OSR) compilation is done for a nested loop: memory operations can be illegally reordered. A regression test that provokes the problem is available at http://hg.openjdk.java.net/hsx/hotspot-comp/hotspot/diff/e3cbc9ddd434/test/compiler/7044738/Test7044738.java.
RFE: 7044738

Area: HotSpot
Synopsis: This problem has been present since JDK 7 build 139 after changes associated with the fix for 5091921, a long standing loop bounds check problem in the loop optimizer. It can happen when a loop's body has complex control flow. The recommended workaround is to specify the `-XX:-UseLoopPredicate -XX:-LoopLimitCheck` flag on the command line.
RFE: 7068051

Area: HotSpot
Synopsis: In JDK 6u14, the behavior of nested references changed subtly because of the bug fix for CR 6684579. This fix addressed a performance anomaly that affected concurrent collector pause time latencies. In JDK 7, the garbage collector continues to behave in an identical manner to JDK 6u14 and subsequent JDK 6 update releases. The issue of nested references is tracked under CR 6990438 and CR 6990442, which will both be addressed in JDK 8.
RFE: 6990438, 6990442, 6684579

Area: HotSpot
Synopsis: Ubuntu 10.10 and newer has a new default security policy that affects Serviceability commands. This policy prevents a process from attaching to another process owned by the same UID if the target process is not a descendant of the attaching process.
The new security policy affects the following components:

- Serviceability Agent (SA)
- `jinfo`
- `jmap`
- `jstack -F`

You can revert to the more permissive policy in one of two ways: 1) temporarily (until the next reboot) by changing the value to 0 in `/proc/sys/kernel/yama/ptrace_scope`, or 2) more permanently by changing the value to 0 in `/etc/sysctl.d/10-ptrace.conf` and rebooting. Option 1 does not require a reboot and option 2 does require a reboot.
RFE: 7050524

Area: HotSpot
Synopsis: In JDK 7 FCS (b147), Hotspot crashes with a `segv` while running `PorterStemmer`, which is part of Apache Lucerne.
The recommended workaround is to specify `-XX:-UseLoopPredicate` on the command line.
The problem has existed at least since JDK 6 Update 23 and JDK 7 build 102 when using `-XX:+AggressiveOpts`, but was exposed with no command line flags in JDK 7 build 134 when the code which

treats the length of a constant `String` as constant was enabled by default as part of the fix for 6942326.
An interim fix for this issue was pushed into the OpenJDK Hotspot source base for Hotspot 22 on July 27, 2011 under 7070134: those who wish to examine or use it can find the changeset at http://hg.openjdk.java.net/hsx/hotspot-comp/hotspot/rev/4e761e7e6e12.
RFE: 6942326, 7070134

Area: Installer
Synopsis: If the user cancels out of the custom "Files in Use" dialog while uninstalling the JRE, it may no longer possible to perform the uninstall though no error is displayed. When this occurs, the `MSI*.log` file created in the `%TEMP%` folder contains the following:

```
MSI (s) (*:*) [*:*:*:*]: Product: Java(TM) 7 -- Removal failed.
MSI (s) (*:*) [*:*:*:*]: Windows Installer removed the product.
  Product Name: Java(TM) 7. Product Version: 7.0.0.
  Product Language: 1033. Removal success or error status: 1602.
```

To work around this problem, delete the `HKEY_CURRENT_USER\Software\JavaSoft\FIUCancel` registry key, if it exists, and retry the uninstallation.

Area: Webstart
Synopsis: JNLP install hints are ignored in some cases. With the default setting (**Install**, if hinted) an application or applet will still not be installed if either:
The JNLP file has no `href` tag for the main JNLP file and `offline-allowed` is not specified.
The platform doesn't support shortcuts.

RFE: 7046670

Area: Plugin
Synopsis: The Java Plugin and Browser use different `https` sessions to perform authentication. Therefore, a username and password needs to be given to the browser and then another username and password needs to be given to Java Plugin.
The Java plugin implementation has been modified to avoid the double authentication for Firefox, but the same solution was not possible for Internet Explorer because Microsoft does not expose the necessary API.
There is a workaround to avoid the double authentication dialog: When the username/password dialog box for Java Plugin appears, there is an option to remember the username and password. Clicking the check box in the dialog causes the plugin to remember the username and password for that particular site, and the authentication dialog will not pop up again.
RFE: 6936012

Area: Plugin
Synopsis: Under Internet Explorer, an applet cannot open a document using the `showDocument` API when the window target name does not follow naming standards, for example if it contains spaces. The HTML specification on the W3C site defines the proper format of names. Excerpted from that site: *ID and NAME tokens must begin with a letter ([A-Za-z]) and may be followed by any number of letters, digits ([0-9]), hyphens ("-"), underscores ("_"), colons (":"), and periods (".").* For example, the following won't work:

```
    showDocument("http://myhost/myDoc.html", "my window");   // Incorrect
```

The following code snippet, which uses a window target name that complies with the naming standard, works:

```
    showDocument("http://myhost/myDoc.html", "my_window");   // Correct
```

RFE: 7074254

Area: Security
Synopsis: The SSLv2Hello handshake protocol, which was used by SSLv3 server implementations to communicate with older SSLv2 server implementations that did not understand SSLv3, is now disabled by default. A side effect of this is that the SSL/TLS extensions are no longer stripped from the hello message. In most cases, this is not a problem because an SSL/TLS peer is supposed to ignore any extensions that it does not understand. However, there may be older server implementations that experience problems.
RFE: 4873188, 6916074

Area: Security

Synopsis: Public key certificates containing Elliptic Curve Cryptography (ECC) keys are not parsed correctly by the SunPKCS11 JCE security provider on Solaris 11. A known workaround is to use the SunEC JCE security provider instead.
RFE: 7054637

Area: Internationalization
Synopsis: Due to a change for CR 4700857, a `GregorianCalendar` instance created by invoking the `toGregorianCalendar()` method on `javax.xml.datatype.XMLGregorianCalendar` may return an incorrect first day of the week when the locale format and location are set to a different language than the system locale on Windows. The following cases are examples of locale format/location being set to a different OS Locale with a different first day of the week. For example:

OS	Format/Location
en_US	de_DE
en_CA	de_DE
de_DE	en_CA
iw_IL	fi_FI, etc

RFE: 4700857

Area: Internationalization
Synopsis: On Linux distributions, pressing keys so that they auto-repeat might hang when using the SCIM/iBus-based input methods in Chinese/Japanese/Korean locales. If this happens, the input method daemon needs to be restarted. Alternatively, use input methods other than the SCIM/iBus-based input methods to avoid this issue entirely.
RFE: 6506617

Area: JSSE
Synopsis: In the SunJSSE provider, Elliptic Curve Cryptography (ECC) Algorithms and Advanced Encryption Standard (AES) based cipher suites are most preferred in JDK 7. (See The SunJSSE Provider in Java Cryptography Architecture Oracle Providers Documentation.) However, some flawed operating system native PKCS11 implementations cannot work well under heavy load as this will cause TLS handshaking failure when the underlying JCE provider is PKCS11 based. It is recommended to upgrade the native PKCS11 libraries, or use one of the following workarounds: a) disable the AES algorithm in the SunJSSE provider, or b) disable the PKCS11 JCE provider in the entire Java implementation.
RFE: 6916074

Area: API: JMX
Synopsis: The `MLet` classloader is unable to load resource files that are contained in jar files. This bug is valid in the JDK 6 and JDK 7 releases. The workaround is to obtain the jar files from the classpath.
RFE: 7055240

Area: API: AWT
Synopsis: Translucent windows support has been added to AWT. Before enabling visual effects by invoking the `Windows.setOpacity()`, `Windows.setShape()`, or `Windows.setBackground()` methods, a developer should verify whether the desired effects are supported. This is done by invoking the `GraphicsDevice.isWindowTranslucencySupported()` method and specifying the desired effect as an argument. If the method reports an effect as supported, enabling it won't throw an exception. However, there are several issues when using the effects on X11 systems (Linux, Solaris).
Per-pixel translucency (`Windows.setBackground()`): while the effect may be reported as supported, windows with this effect enabled may not look transparent visually because the system isn't running any compositing manager. The system should either be running a compositing window manager (such as compiz), or a separate compositing manager (such as the xcompmgr). Unfortunately, there is no way to detect whether a compositing manager is currently running, so Java is unable to report whether the transparent window will actually appear transparent. It can only check and report that enabling the effect for a window is supported by the native platform.
Simple translucency (`Window.setOpacity()`): the effect may be reported as unavailable in Java, while the window manager actually supports it for active applications and is able to display semi-transparent windows. This happens because a window manager (e.g. some versions of Metacity) doesn't report the effect as supported.

Please note that this is not a defect in Java, but rather a bug in the window manager. More details are available in CR 6762511.

RFE: 6990921

Area: API: AWT
Synopsis: This issue is specific to `Window.toFront()` and `Window.toBack()` behavior on X11 systems (Linux, Solaris). AWT always performs all necessary actions required to bring a window to the top or the bottom in the stacking order, according to the specifications of the windowing systems (e.g. the ICCCM and/or the EWMH for X11 systems). Whether the actions take effect or not depends on the window manager currently running in the system. Some windowing systems enforce additional constraints on when a window may be brought to the top in the window stacking order. In most cases, the purpose for such additional constraints is to enhance usability or security of these windowing environments. In other cases, these may simply be bugs in the windowing systems. This means that invoking the `Window.toFront()` or `Window.toBack()` methods may not always produce the expected results.
While this issue is, to some extent, reflected in the specification of the `toFront()` method, we've received several user complaints on this particular behavior while developing JDK7. All of these CRs have been closed as "Not a Defect" because AWT can't short-circuit the additional constraints, or work around any bugs in third-party software. If the problem is encountered with a Java application, in most cases it makes sense to check if native applications behave the same way, and file a bug against the windowing system accordingly.

Area: API: 2D
Synopsis: Due to a bug in the platform's `/usr/lib/libXrender.so` on older Linux versions, the new Java 2D Xrender pipeline may cause the JRE to hang, or exit without warning due to the Xserver disconnecting the application. The bug is documented on freedesktop.org, and affects applications that use the `GradientPaint` class. There is no known workaround other than to avoid use of gradients in your application. This bug is known to affect Red Hat Enterprise Linux (RHEL) 5.6 and earlier and Oracle Enterprise Linux (OEL) 5.6 and earlier. Other versions of a similar vintage may be similarly affected. The solution is to install version 0.9.3 or later of the Xrender library, or upgrade to a later Linux distribution, such as OEL 6. The Xrender pipeline is disabled by default in JDK 7. It can be enabled by using the following syntax for the system property: "`-Dsun.java2d.xrender=True`". This will issue a warning about this Xrender bug if the implementation cannot verify that the bug is not present.
For more information on the Xrender pipeline, see Java 2D Enhancements in Java SE 7.
RFE: 7032904

Area: API: Text
Synopsis: Text attributes are checked in to the `Bidi(AttributedCharacterIterator)` constructor before the iteration index is initialized. If the index isn't set to the first char, the `Bidi.baseIsLeftToRight()` method may return a wrong value. To work around this, invoke the `AttributedCharacterIterator.first()` method to reset the iteration index before calling the `Bidi(AttributedCharacterIterator)` constructor if the iteration index is out of the text range.
RFE: 7042148

Area: Core Libs/`java.net`
Synopsis: On Windows platforms, the IPv4 stack needs to be configured in addition to the IPv6 stack, for the IPv6 stack to work. For more information on IPv6 configuration, see Networking IPv6 User Guide.
Bug: 8040229