



- Installing
- Contributing
- Sponsoring
- Developers' Guide
- Vulnerabilities
- JDK GA/EA Builds
- Mailing lists
- Wiki · IRC
- Bylaws · Census
- Legal
- Workshop
- JEP Process
- Source code
- Mercurial
- GitHub
- Tools
- Git
- jtreg harness
- Groups
- (overview)
- Adoption
- Build
- Client Libraries
- Compatibility & Specification
- Review
- Compiler
- Conformance
- Core Libraries
- Governing Board
- HotSpot
- IDE Tooling & Support
- Internationalization
- JMX
- Members
- Networking
- Porters
- Quality
- Security
- Serviceability
- Vulnerability
- Web
- Projects
- (overview, archive)
- Amber
- Audio Engine
- CRaC
- Caciocavallo
- Closures
- Code Tools
- Coin
- Common VM Interface
- Compiler Grammar
- Detroit
- Developers' Guide
- Device I/O
- Duke
- Font Scaler
- Galahad
- Graal
- Graphics Rasterizer
- IcedTea
- JDK 7
- JDK 8
- JDK 8 Updates
- JDK 9
- JDK (... , 21, 22)
- JDK Updates
- JavaDoc.Next
- Jigsaw
- Kona
- Kulla
- Lambda
- Lanai
- Leyden
- Lilliput
- Locale Enhancement
- Loom
- Memory Model
- Update
- Metropolis
- Mission Control
- Modules
- Multi-Language VM
- Nashorn
- New I/O
- OpenJFX
- Panama
- Penrose
- Port: AArch32
- Port: AArch64
- Port: BSD
- Port: Haiku
- Port: Mac OS X
- Port: MIPS
- Port: Mobile
- Port: PowerPC/AIX
- Port: RISC-V
- Port: s390x
- Portola
- SCTP
- Shenandoah
- Skara
- Sumatra
- Tiered Attribution
- Tsan
- Type Annotations
- Valhalla
- Verona
- VisualVM
- Wakefield
- Zero
- ZGC



JEP 449: Deprecate the Windows 32-bit x86 Port for Removal

<i>Author</i>	George Adams & Bruno Borges
<i>Owner</i>	George Adams
<i>Type</i>	Feature
<i>Scope</i>	Implementation
<i>Status</i>	Closed / Delivered
<i>Release</i>	21
<i>Component</i>	hotspot/other
<i>Discussion</i>	jdk dash dev at openjdk dot org
<i>Effort</i>	S
<i>Duration</i>	S
<i>Reviewed by</i>	Jesper Wilhelmsson
<i>Endorsed by</i>	Dalibor Topic, Vladimir Kozlov
<i>Created</i>	2023/02/24 15:30
<i>Updated</i>	2023/08/18 18:03
<i>Issue</i>	8303167

Summary

Deprecate the Windows 32-bit x86 port, with the intent to remove it in a future release.

Goals

- Update the build system to issue an error message when an attempt is made to configure a build for Windows 32-bit x86 (x86-32). The error message will be suppressible via a new configuration option.
- Mark the port, and related port-specific features, as deprecated for removal in the relevant documentation.

Non-Goals

- It is not a goal to change the status of the affected port in any prior release. The earliest release to which this JEP could be targeted is JDK 21.
- It is not a goal to deprecate any other 32-bit port.

Motivation

- Allow contributors in the OpenJDK Community to accelerate the development of new features and enhancements that will move the platform forward.
- The implementation of [JEP 436 \(Virtual Threads\)](#) for Windows x86-32 falls back to the use of kernel threads and therefore does not bring the expected benefits of Project Loom.
- Windows 10, the last Windows operating system to support 32-bit operation, will reach End of Life in [October 2025](#).

Description

An attempt to configure a Windows x86-32 build will produce the following output:

```
$ bash ./configure
...
checking compilation type... native
configure: error: The Windows 32-bit x86 port is deprecated and may be removed in a future release. \
Use --enable-deprecated-ports=yes to suppress this error.
configure exiting with result code 1
$
```

The new build configuration option `--enable-deprecated-ports=yes` will suppress the error and continue:

```
$ bash ./configure --enable-deprecated-ports=yes
...
checking compilation type... native
configure: WARNING: The Windows 32-bit x86 port is deprecated and may be removed in a future release.
...
Build performance summary:
* Cores to use:      32
* Memory limit:     96601 MB
```

```
The following warnings were produced. Repeated here for convenience:
WARNING: The Windows 32-bit x86 port is deprecated and may be removed in a future release.
$
```

Alternatives

The alternative would be to continue to support Windows x86-32. This would require active maintainers who can provide a sustainable and performant implementation of Virtual Threads, as well as future JEPs, to ensure that the JDK on Windows x86-32 continues to meet the expectations of Java developers.

Risks and Assumptions

32-bit JVMs are still used on Windows due to integration with 32-bit native libraries (DLLs). Their users cannot migrate directly to 64-bit JVMs because a 64-bit process on Windows cannot load 32-bit DLLs. While Windows x64 is capable of running 32-bit applications by emulating an 32-bit environment through [WOW64](#), applications will suffer [dramatic performance degradation](#) despite the assumed memory footprint benefits.

We therefore assume that

- Users can continue to run existing builds of the Windows 32-bit JVM to integrate with native 32-bit libraries and, if necessary, expose 32-bit functionality via remote APIs to be consumed by applications running on a 64-bit JVM within the same environment; and
- Legacy systems are unlikely to migrate to versions of the JDK following the release of Java 21.