

Oracle Jipher 20 Release Notes

- [Oracle Jipher 20](#)

Oracle Jipher 20

Oracle Jipher is a Java Cryptographic Service Provider (CSP) that dynamically loads a Federal Information Processing Standards (FIPS) 140-3 validated OpenSSL cryptographic module at runtime and uses it as the underlying cryptographic implementation. It enables the deployment of Java applications in FIPS-regulated environments. Jipher makes its cryptographic services available to Java developers using the Java Cryptography Architecture (JCA).

Oracle Jipher is offered as part of the [Oracle Java SE Universal Subscription](#) and to Oracle customers running Java workloads in Oracle Cloud Infrastructure (OCI). Note that the Jipher feature included in the Java SE Subscription has a shorter support lifecycle than the corresponding supported JDK feature versions. Please refer to the [Java SE Support Roadmap](#) for more information.

Supported Runtimes

- Oracle JDK 27
- Oracle JDK 25
- Oracle JDK 21
- Oracle JDK 17

Supported Platforms

- Oracle Linux 10, 9, and 8 on x86-64 and aarch64
- Red Hat Enterprise Linux (RHEL) 10, 9, and 8 on x86-64 and aarch64
- macOS 26 on M series Apple silicon
- Windows Server 2025

Important Changes and Information

- **Runtime Native Library Dependencies**

In Jipher 10.x, Jipher's runtime native library dependencies are embedded in the `jipher-jce` JAR. At runtime, the native libraries appropriate for the platform on which Jipher is deployed are automatically extracted from the `jipher-jce` JAR file to a temporary directory in the file system and loaded from there into the JVM process.

In Jipher 20, Jipher's runtime native library dependencies are not embedded in the `jipher-jce` JAR. They are distributed in a separate `jipher-native` package. The native library dependencies appropriate for the platform on which Jipher is deployed must be unpacked from the `jipher-native` package to the file system before the `jipher-jce` JAR is loaded by a JVM.

See Jipher Deployment for more information.

- **Java Module System**

The library is now delivered as an explicit module rather than an automatic module. The `com.oracle.jipher` module does not export any packages.

Applications that list the `jipher-jce` JAR on the module path can no longer create an instance of the `com.oracle.jipher.provider.JipherJCE` class directly or through reflection:

```
import com.oracle.jipher.provider.JipherJCE;
...
// Cannot create an instance of JipherJCE directly
Provider provider = new com.oracle.jipher.provider.JipherJCE();

// Cannot create an instance of JipherJCE through reflection
Class<?> jipherClass =
Class.forName("com.oracle.jipher.provider.JipherJCE");
Provider provider = (Provider)
jipherClass.getDeclaredConstructor().newInstance();
```

An instance of the `JipherJCE` provider can instead be created using the service loader:

```
Provider provider = ServiceLoader.load(Provider.class).stream()
    .filter(provider ->
provider.type().getName().equals("com.oracle.jipher.provider.JipherJCE"))
    .map(ServiceLoader.Provider::get)
    .findFirst()
    .orElseThrow(() -> new IllegalStateException("JipherJCE
provider not found"));
```

- **The Jipher Provider is Statically Registered by its Provider Name, not its Class Name**

Applications that list the `jipher-jce` JAR on the module path can no longer statically register the JipherJCE provider using its class name in the security properties file (which is, by default `$JAVA_HOME/conf/security/java.security`; see [The Security Properties File](#) in *Java Platform, Standard Edition Security Developer's Guide*):

```
# Invalid: Cannot register the JipherJCE provider using its class
name
security.provider.1=com.oracle.jipher.provider.JipherJCE
```

Applications should statically register the JipherJCE provider using its provider name, for example:

```
security.provider.1=JipherJCE
```

See [Registering Jipher as the Most Preferred Provider](#) for more information.

- **Support for Static Methods `isAvailable` and `loadingException`**

The static methods `JipherJCE.isAvailable()` and `JipherJCE.loadingException()`, which were present in Jipher 10.x, have been removed in Jipher 20.

In Jipher 10.x, an application could use these methods to determine whether OpenSSL was available and, if not, why OpenSSL could not be loaded.

In Jipher 20, an application can determine whether OpenSSL is available by attempting to load JipherJCE with `ServiceLoader`. If the load succeeds, then OpenSSL is available. If it throws a `ServiceConfigurationError`, then the reason OpenSSL could not be loaded is available from the final `Throwable` in the chain of exceptions:

```
try {
    ServiceLoader.load(Provider.class).stream()
        .filter(provider ->
            provider.type().getName().equals("com.oracle.jipher.provider.JipherJCE"))
        .map(ServiceLoader.Provider::get)
        .findFirst()
        .orElseThrow(() -> new IllegalStateException("JipherJCE
provider not found"));
} catch (ServiceConfigurationError sce) {
    if (sce.getCause() instanceof ProviderException
        providerException) {
        providerException.printStackTrace();
    }
}
```

See [Reporting Misconfiguration](#) for more information.

- **Addition of Support for Post-Quantum Cryptographic (PQC) Algorithms**

Jipher 20 adds support for Module-Lattice (ML) based algorithms:

- Key Encapsulation Mechanisms (KEM): ML-KEM, ML-KEM-512, ML-KEM-1024
- Digital signature algorithms (DSA): ML-DSA-44, ML-DSA-65, ML-DSA-87

KeyFactory, KeyPairGenerator, KEM, and Signature service support is provided.

See Supported Algorithm Strings for more information.

- **Addition of Support for Key Derivation Function (KDF) Service Algorithms**

Jipher 20 adds support for HMAC-based Key Derivation Function (HKDF) KDF service algorithms.

See Supported Algorithm Strings for more information.

- **Removal of Support for HmacPBE* Mac Algorithms**

By default, Jipher 20 does not provide HmacPBE* Mac algorithms (see [Mac Algorithms](#) in *Java Security Standard Algorithm Names*), as defined in [Appendix B.4: Keys for Password Integrity Mode](#) in RFC 7292. These password-based MAC algorithms are not allowed by FIPS 140-3.

On JDK 25 and earlier, the SUN provider's PKCS12 KeyStore service employs HmacPBE* Mac algorithms for the integrity check when reading KeyStore instances from or writing them to persistent storage. If your application absolutely must load a PKCS12 KeyStore from or store one to persistent storage, then you can set the system property `jipher.nonfips.supportPkcs12Kdf` to `true` to configure Jipher to provide support for HmacPBE* Mac algorithms. This setting deviates from the FIPS 140-3 standard.

See Optionally Supported Non-FIPS 140-3 Allowed Algorithms and Configuring Jipher Through System and Security Properties for more information.

- **Removal of Support for the PBESWithSHA1AndDESede Algorithm**

By default, Jipher 20 does not provide the PBESWithSHA1AndDESede algorithm. This password-based algorithm is not allowed by FIPS 140-3. Jipher can be configured to provide support for the PBESWithSHA1AndDESede algorithm by setting the system property `jipher.nonfips.supportPkcs12Kdf` to `true`.

See Optionally Supported Non-FIPS 140-3 Allowed Algorithms and Configuring Jipher Through System and Security Properties for more information.

Additional Information

- [Oracle Jipher Documentation](#)
- Download Jipher from [Java Verified Portfolio](#)
- For support, refer to [My Oracle Support](#)

Oracle Jipher Release Notes, Release 20
G59618-01

Copyright © 2025, 2026, Oracle and/or its affiliates

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.