Orchestrating a brighter world

NEC

JDK8からJDK11に移行する際の注意点について

2020年8月17日　第1.2版

日本電気株式会社
クラウドプラットフォーム事業部

目次

はじめに

本書に関する注意

1. WebOTXは日本電気株式会社の登録商標です。
2. OracleとJavaは、Oracle Corporation 及びその子会社、関連会社の米国及びその他の国における登録商標です。
3. 本書の一部または全部を無断に転載することを禁じます。
4. 本書に関しては将来予告なしに変更することがあります。
5. 弊社の許可なく複製、改変することを禁じます。
6. 対処した結果の影響については、弊社は一切責任を負いません。

背景と目的

背景

2019年３月に、Java SE Development Kit 11(JDK 11)に対応した WebOTX Application Server V10.2 をリリースしました。 Oracle社によれば、 Java SE Development Kit 8(Oracle JDK 8)の公式アップデートの提供は、個人ユーザには2020年12月まで、商用ユーザには2019年1月までで終了します。それ以降は、Oracle JDK 8 で不具合が発覚したとしても修正されることはありません。

そのため、Oracle JDK 8 で動作するシステムは、Oracle社が提供する Oracle Java Subscription(有償) または Oracle社がビルドしたOpenJDK (無償) に切り替える必要があります。

Oracle Java Subscription(有償) の JDK 8 以降のリリースで長期サポート(Long-Term-Support:LTS)が表明されている JDK 11 に切り替える場合、JDK 8 と JDK 11 の間では、大きく互換性を損ねる変更が行われているため、移行の際には注意が必要です。

目的と対象読者

本資料では、アプリケーションの移行や運用管理における注意点について記載しています。WebOTX Application Server V9系、または V10.1系 の利用者が、V10.2系 以降の導入を契機にアプリケーションの動作環境を JDK11 に変更する場合に有用です。

表記ルール

パスの表記

本資料では、パスの表記を ”/” としています。Windows系の場合は ”/” を ”¥” に読み替えてください。

インストールディレクトリの表記

- ${AS_INSTALL}・・・WebOTXのインストールディレクトリ
- ${INSTANCE_ROOT}・・・WebOTXのドメインディレクトリ
- ${JAVA_HOME}・・・Java のインストールディレクトリ

JDK 8 と JDK 11 の非互換項目

java.io.FilePermission がファイルパスを正規化しなくなった

現象

JDK 9 ではパフォーマンスの向上を目的として、java.io.FilePermission オブジェクト構築時に指定したファイルパスを正規化しないように変更されました。これは以下のようなファイルパスが、同じファイルとみなされないことを意味します。

その結果、FilePermission の権限が不足してエラーが発生する場合があります。

- 絶対パスと相対パス
- シンボリックリンクとそのターゲット
- Windowsで表示されるパスとDOS 形式の 8.3 名を使用したパス
  (例 “C：¥ Program Files”と “C：¥ PROGRA～1”)

対処

以下、何れかの対処が必要となります。

- システムプロパティ jdk.io.permissionsUseCanonicalPath に true を設定してください。旧互換の動作になります。
- Java のポリシーファイルに、実装で指定している内容と異なる表記でパスを指定している場合は、同じパス表記に変更してください。

rt.jar と tools.jar が削除された

現象

JDK 9 では JDK/JRE のライブラリ構成が変更されました。従来、${JAVA_HOME}/lib に配置されていた rt.jar や tools.jar は特殊なファイルとして格納されています。jar ファイルの形式では存在しません。クラスパスの指定において、存在しない jar ファイルは無視されるため、指定していること自体に問題はありませんが、tools.jar の有無によって処理を分けている場合は、意図した動作にならない可能性があります。

対処

- スクリプトやプログラム内で rt.jar や tools.jar の存在有無により処理を分けている場合は、修正を行ってください。

アプリケーションでの使用例

```
File toolsJar = new File(jdkDir + "/lib/tools.jar" );
if (toolsJar != null && toolsJar.exists()) {
    toolsJarPath = toolsJar.getPath();
}
```

上記の toolsJar.exists() は常に false になります。

lib/ext が廃止された

現象

JDK 9 で拡張機能メカニズムが削除されたことにより、${INSTANCE_ROOT}/lib/ext に配置されたライブラリはロードされなくなりました。そのため、クラスパス設定の手段として ${INSTANCE_ROOT}/lib/ext を利用することはできません。

対処

- ${INSTANCE_ROOT}/lib/ext への配置以外の方法で、クラスパスを設定してください。

例

JDBCドライバを ${INSTANCE_ROOT}/lib/ext に配置していた場合、対処として以下のような方法があります。

- 対処1
  - JDBCドライバを ${INSTANCE_ROOT}/lib に配置してください。
- 対処2
  - JDBCドライバを任意のパス (例えば D:¥ojdbc8.jar) に配置し、以下のように server-classpath を設定してください。

```
otxadmin>set server.java-config.server-classpath=D:¥¥ojdbc8.jar
```

内部APIが隠蔽された

現象

JDK 9 で JDK の一部の内部実装が隠蔽されました。そのため、隠蔽された内部 API を使用している場合、JDK 11 を使用したコンパイルに失敗します。また、JDK 8 以下でコンパイルしたものを JDK 11 で実行した場合、実行時に隠蔽された内部APIが見つからず java.lang.ClassNotFoundException が発生します。

対処

- JDK 11 に付属する jdeps コマンドを使用し、隠蔽された内部 API を使用しているか確認します。jdeps により代替 API が示された場合は、その API を使用するよう修正してください。代替 API が存在しない場合は、実装の見直しが必要となります。

jdeps の実行例

```
${JAVA_HOME}/bin/jdeps <jarファイル or ディレクトリ>
```

モジュールが削除された

現象

- JDK 9 では、下記のモジュールがデフォルトでは解決されなくなりました。さらに、JDK 11ではこれらのモジュールが削除されました。この結果、これらのクラスを使用するアプリケーションのコンパイル時にクラスが見つからないというエラーが発生します。また、実行時には java.lang.NoClassDefFoundError が発生します。

モジュール名	対応するパッケージ名
java.activation	javax.activation
java.corba	javax.activity, javax.rmi, javax.rmi.CORBA, org.omg.*
java.transaction	javax.transaction
java.xml.bind	javax.xml.bind.*
java.xml.ws	javax.jws, javax.jws.soap, javax.xml.soap, javax.xml.ws.*
java.xml.ws.annotation	javax.annotation

対処

- 削除されたモジュールはJava EEに含まれます。そのため、WebOTX Application Server上で動作するアプリケーションの場合は、実行時の対処は不要です。
- アプリケーションのビルド時またはWebOTX Application Serverと通信するクライアントアプリケーションの実行時には、WebOTXマニュアルのアプリケーション開発ガイドを参照して指定されたWebOTXのjarファイルをクラスパスに追加してください。あるいはMaven Central Repositoryから上記パッケージを含むjarファイルとその依存ライブラリをダウンロードしてクラスパスに追加してください。

JavaDBが同梱されなくなった

現象

- JDK 8 Update 181 以降、および JDK 9 以降で JavaDB が同梱されなくなりました。

対処

- JavaDB が必要な場合は、個別に Apache Derby （http://db.apache.org/derby/) をインストールしてください。

java.version 表記が変更された

現象

- システムプロパティ java.version が示すバージョン番号の表記 (System.getProperty(“java.version”); の戻り値) は、JDK 9 で以下のように変更されました。

  - JDK 8 まで

    jdk1.8.0_92 の場合 **1.8.0_92**

  - JDK 9 以降

    jdk9.0.0.0 の場合 **9**

    jdk9.0.1.0 の場合 **9.0.1**

    jdk11.0.2.0 の場合 **11.0.2**

対処

- System.getProperty(“java.version”); の戻り値に対して、JDK 8 の形式を想定した処理を行っている場合は、 JDK 9 以降の形式を想定した処理に修正してください。

java.home の指すパスが変更された

現象

- システムプロパティ java.home が指すパス (System.getProperty("java.home"); の戻り値) は、JDK 9 で以下のように変更されました。

  - JDK 8 まで

    JDK インストールディレクトリの jre ディレクトリ

    例) C:¥Program Files¥Java¥jdk1.8.0_92¥jre

  - JDK 9 以降

    JDK インストールディレクトリ

    例) C:¥Program Files¥Java¥jdk-11

対処

- System.getProperty(“java.home”);などを使用していた場合、JDK 8 とJDK 9 以降で取得結果が異なります。処理を見直してください。

リモートデバッグの既定の動作が変更された

現象

- JDK 9 でリモートデバッグ用の –Xrunjdwp オプションは、既定でローカルマシンからしかプロセスにアタッチできないよう変更されました。

対処

- リモートマシンからアタッチする場合、デバッグオプションを以下のように変更してください。

【修正前】

```
-Xdebug -Xrunjdwp:transport=dt_socket,server=y,suspend=y,address=4004
```

【修正後】

```
-Xdebug -Xrunjdwp:transport=dt_socket,server=y,suspend=y,address=*:4004
```

参考情報

参考情報

- 「Java Platform, Standard Edition　~ Migrating to JDK 9 ~」
  https://docs.oracle.com/javase/9/migrate/toc.htm#JSMIG-GUID-7744EF96-5899-4FB2-B34E-86D49B2E89B6

改版履歴

版数	発行日	改版内容
1.0	2019/8/23	新規作成
1.1	2019/12/3	体裁を変更
1.2	2020/8/17	「モジュールが削除された」のページの「対処」を修正


Orchestrating a brighter world
NEC