

Dash0 と R8 の統合について



目次

I.	はじめに.....	2
	(1)公式マニュアルについて	2
II.	R8 の登場	2
	(1) R8 の特徴について	2
	(2) R8 の登場によって DashO の変更点	3
	(3) R8 環境における DashO の役割.....	3
III.	DashO と R8 設定方法	4
	(1) DashO GUI を実行する	4
	(2) [Next]をクリックして、[Select Application Type]ページに移動します.....	5
	(3) [Android (Most projects)]を選択し、[Next]をクリックします。.....	5
	(4) app の build.gradle があるディレクトリを選択し、[Next]をクリックします。.....	6
	(5) Gradle ビルド構成で変更される内容の概要を読み、[Finish]をクリックします。...	6
	(6) プロジェクトをロードし、ビルドする.....	7
	(7) DashO 設定ファイル(project.dox)の確認.....	8
IV.	DashO の機能の設定方法	9
	(1) 制御フローの難読化.....	9
	(2) 文字列の暗号化	10

I. はじめに

AndroidStudio 3.4 からコードシュリンカーに R8 が標準で使われるようになりました。

R8 が使われることにより今までの仕組みでは DashO が使えなくなったため、大きく改良されました。DashO 9.3 より前のバージョンをお使いのユーザー様は、利用方法に戸惑うことが予想されますので、この資料では、R8 の概要や、DashO と R8 の統合方法についてご説明します。

(1) 公式マニュアルについて

DashO の公式マニュアルでは、DashO を使用する上での詳細が記述されています。DashO を使用する上で、より詳しい内容は、WEB ページをご覧ください。

<https://www.agtech.co.jp/download/manual/preemptive/>

II. R8 の登場

R8 は ProGuard の後継として導入された Android 用のコードシュリンカーです。R8 は、Android Gradle plugin 3.4.0 からは標準で有効になりました。DashO は R8 と連携して Android アプリケーションまたはライブラリを保護するようになります。

(1) R8 の特徴について

R8 の特徴としては、以下が挙げられます。

- ・コードの圧縮、desugar、dex をまとめて一度に実行可能。
- ・従来の ProGuard と比較してより速く、出力サイズも小さい。
- ・R8 と ProGuard は互換性があり、ProGuard 設定ファイルをそのまま読める

R8 を有効にすると、名前変更、除去、メソッド呼び出しの削除および最適化が行われます。DashO でも同様の機能を持ち合わせておりますが、処理が重複してしまうため、R8 の変換分を自動的に無効にします。

詳細については、下記の URL をご参照ください。

<https://www.agtech.co.jp/products/preemptive/blog/article02.html>

(2) R8 の登場によって DashO の変更点

DashO は R8 と連携するために設計された、新しい Android モードを用意しました。Android モードは DashO 10.0 からご利用することが可能で、Android 用の新しい DashO Gradle プラグインが付属しています。

この新しいプラグインの特徴は下記になります。

- ・ Gradle により自然に統合します。
- ・ Android モードの DashO プロジェクトファイル (.dox) を自動的に生成できます。
- ・ どのコンポーネントを処理する必要があるかを自動的に識別し、自動的に DashO に渡すため、手作業での入力は不要です

※入力と出力はプラグインによって管理され、R8 によって処理される機能は DashO で無効になっています。

(3) R8 環境における DashO の役割

DashO は、制御フローや文字列暗号化の静的な保護と、デバッグ検知や改ざん検知などの動的な保護を適用します。R8 は、以前使用していたのと同まったく同じ ProGuard 構成ファイルを使用して縮小化と最適化を実行します。

III. DashO と R8 設定方法

新規プロジェクトウィザードを使用して、DashO と R8 の統合手順を説明します。

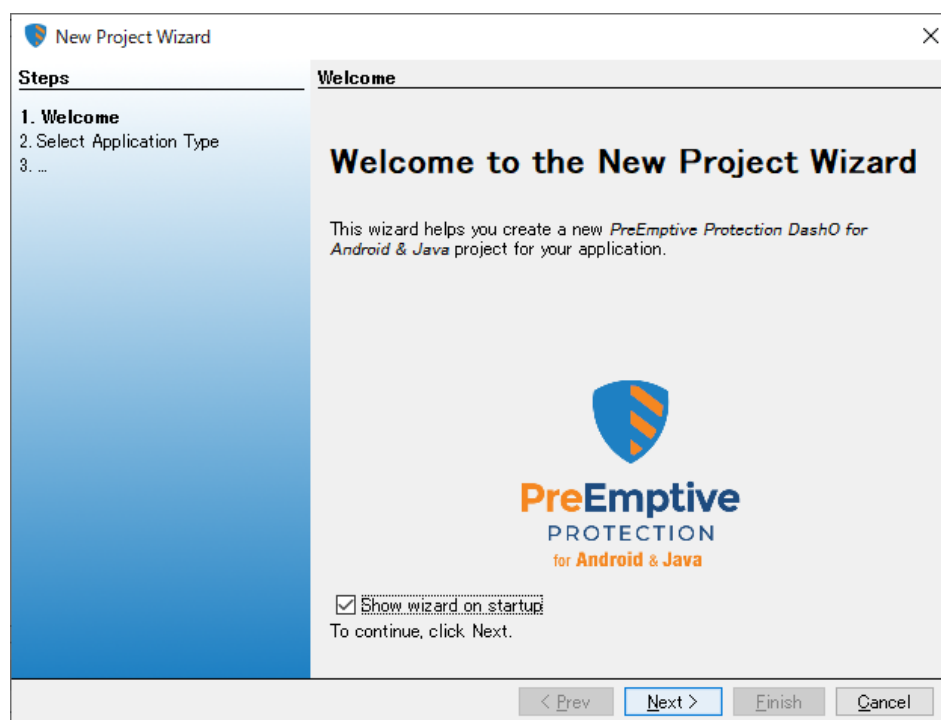
新しい統合を実現するために、DashO をゼロから再考し、UI、ビルド時の動作、DashO を Gradle ビルドに統合する方法に大きな変更を加えた新しい Android モードを使用します。統合がより簡単で、理解と構成がより簡単になります。具体的な手順を 1 ステップずつ紹介します。

※R8 側の設定に関しては今回説明しませんが、下記のドキュメントが参考になります。

<https://r8-docs.preemptive.com/>

(1) DashO GUI を実行する

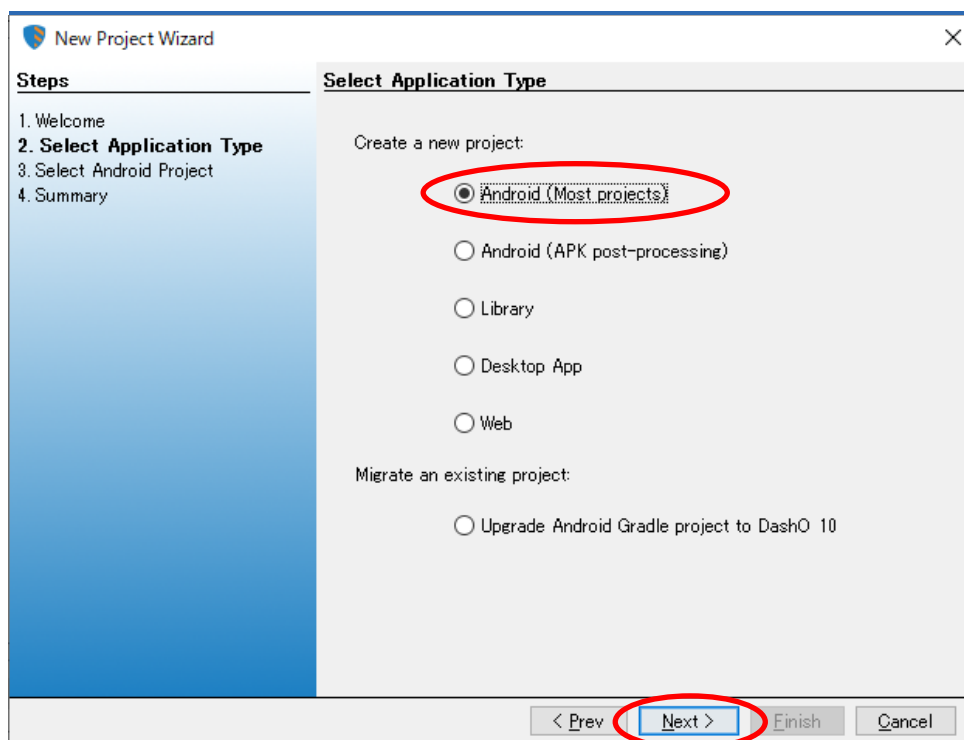
DashO GUI を実行すると、New Project Wizard 画面が立ち上がります。



(2) [Next]をクリックして、[Select Application Type]ページに移動します

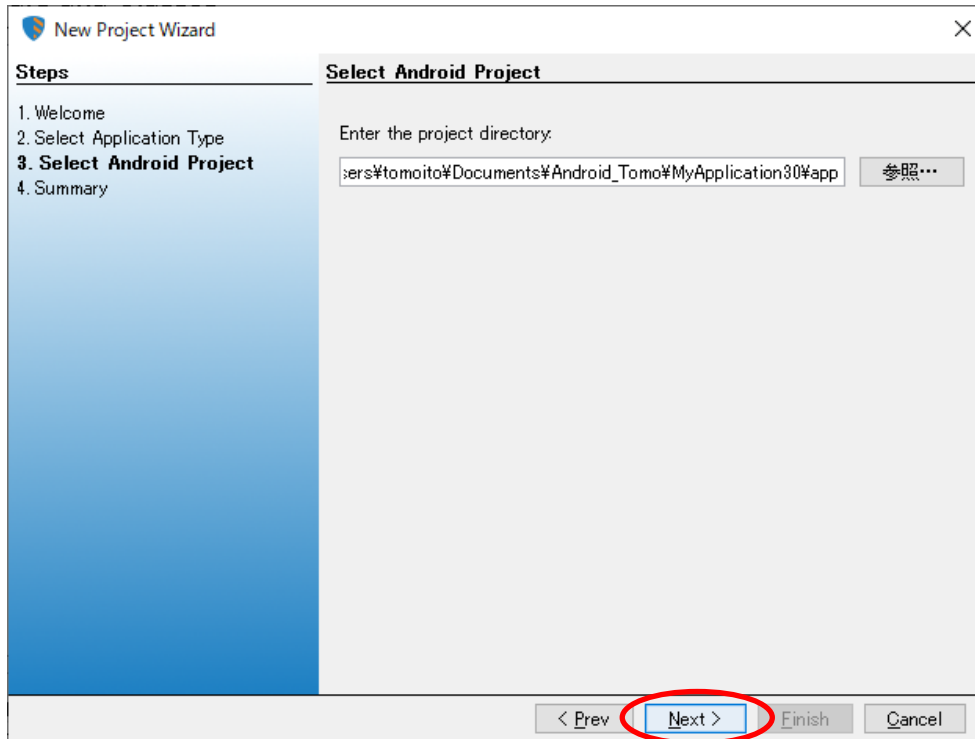


(3) [Android (Most projects)] を選択し、[Next]をクリックします。

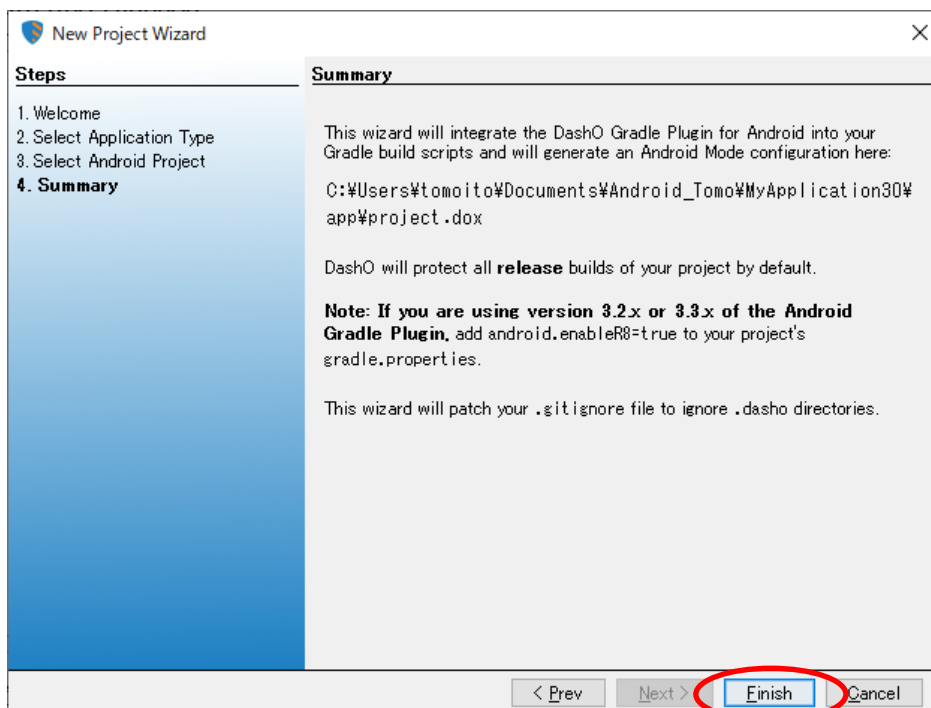


※DashO 10 以前の「An Android Application or library」に該当します。

(4) app の build.gradle があるディレクトリを選択し、[Next]をクリックします。



(5) Gradle ビルド構成で変更される内容の概要を読み、[Finish]をクリックします。

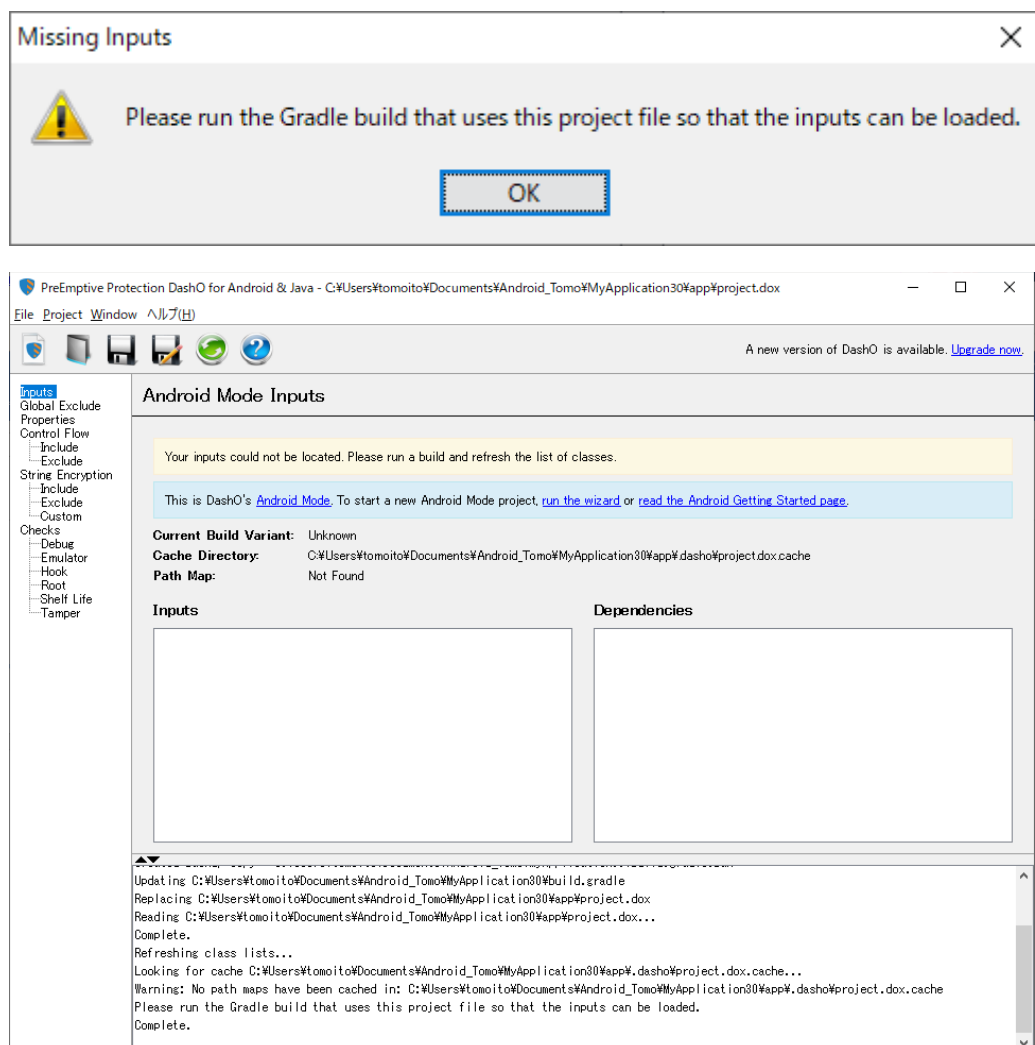


※Android Gradle プラグインのバージョン 3.4 以降では、デフォルトで R8 がシュリンクとして設定されています。Android Gradle プラグインのバージョン 3.3 以前を使用している場合、次のステートメントを gradle.properties ファイルに追記する必要があります。

```
android.enableR8=true
```

(6) プロジェクトをロードし、ビルドする

上記 (5) の手順後、入カクラスの一覧を更新するように求められます



上記のメッセージが表示されたら、gradlew clean build の実行をしてください。

これで、DashO と R8 の統合が完了しました。

※Android アプリを難読化（apk 直接を除く）する場合、DashO GUI では難読化はできません。編集を保存後、gradlew build を実行することで、難読化が実行されます。

(7) DashO 設定ファイル(project.dox)の確認

DashO の設定ファイル (project.dox ファイル) が、app 直下のフォルダに作成されます。DashO 9.3 以前に比べて、項目が少ないですが、下記のような状態がデフォルトになっています。

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<dasho mode="android" version="10.0.0">

    <controlflow option="on"/>

    <stringencrypt option="on"/>

    <injection/>

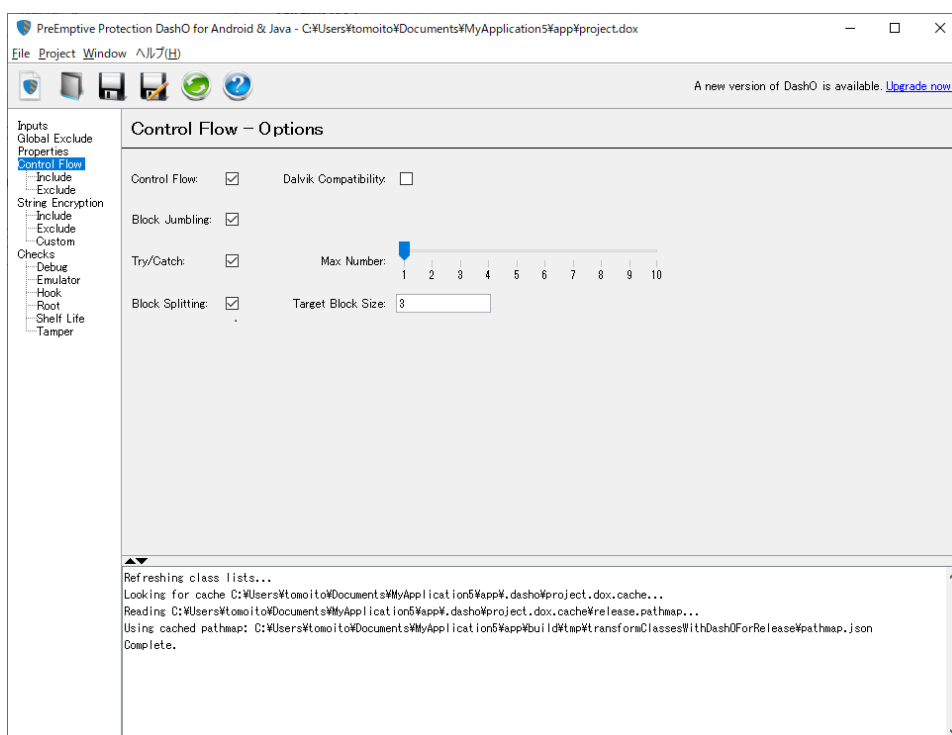
</dasho>
```

IV. DashO の機能の設定方法

DashO が提供する機能はすべて、DashO GUI を使用して構成できます。DashO GUI で設定した内容は、全て project.dox に XML 形式でテキストとして保管されます。また、project.dox を DashO GUI で開き直すことで、設定した内容を変更することが可能です。

この章では、DashO の代表的機能である「制御フロー」、「文字列の暗号化」の設定についてご説明します。

(1) 制御フローの難読化

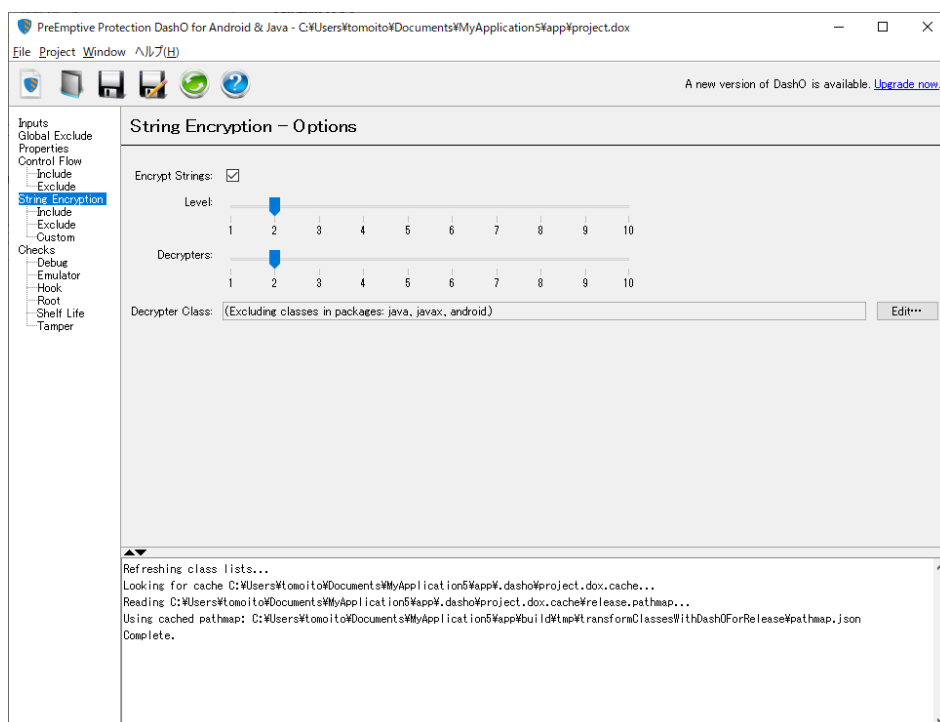


[Control Flow]チェックボックスを使用して、制御フローの難読化を完全に有効または無効にできます。制御フローの難読化が有効になっている場合は、対応するチェックボックスを使用して、Block Jumbling、Try/Catch、Block Splitting を有効または無効にすることもできます。

また、パフォーマンスの問題が懸念される場合は、メソッド、クラスの除外も可能です。詳細は下記をご参照ください。

https://www.preemptive.com/dasho/pro/userguide/en/understanding_obfuscation_control.html#includes-excludes

(2) 文字列の暗号化



[Encrypt Strings]チェックボックスを使用して、文字列の暗号化を完全に有効または無効にできます。文字列の暗号化が有効になっている場合、暗号化技術の強度を制御したり、復号化メソッドを配置する場所を制御したりできます。

制御フローの難読化同様、パフォーマンスの問題が懸念される場合は、メソッド、クラスの除外も可能です。詳細は下記をご参照ください。

https://www.preemptive.com/dasho/pro/userguide/en/ui_main_string_encryption.html#string-encrypt-include-exclude