

VeriX iOS Build & Apple Developer Integration

This document outlines the step-by-step procedure for configuring, building, and deploying the **VeriX** iOS mobile application using **Expo (EAS Build)**, **Apple Developer Account**, and **TestFlight**, following **SOC 2 Least Privilege Access** security principles.

? Executive & Technical Overview

- **App Name:** VeriX
 - **Bundle Identifier:** `com.nexus33.verix`
 - **Build System:** Expo Application Services (EAS Build & Submit)
 - **Target OS:** iOS 14.0+
 - **Distribution Target:** TestFlight (Internal & External Beta Testing)
 - **Security & Compliance:** SOC 2 Type II compliant (Developer permissions strictly scoped; Account Holder handles Identifier registration).
-

?? Section 1: `eas.json` Profile Configuration

The `eas.json` file is configured so that test builds target physical iOS devices via TestFlight using Apple Store distribution (`"distribution": "store"` and `"simulator": false`), while preserving Android APK generation.

```
{
  "expo": {
    "name": "VeriX",
    "slug": "nexus33-app-verix",
    "version": "1.0.9",
    "orientation": "portrait",
    "icon": "./assets/images/icon-ios.png",
    "userInterfaceStyle": "dark",
    "scheme": "verix",
```

```
"ios": {
  "supportsTablet": true,
  "bundleIdentifier": "com.nexus33.verix",
  "buildNumber": "1",
  "infoPlist": {
    "ITSAppUsesNonExemptEncryption": false
  }
},
"android": {
  "adaptiveIcon": {
    "foregroundImage": "./assets/android-icon-foreground.png",
    "backgroundImage": "./assets/android-icon-background.png",
    "monochromeImage": "./assets/android-icon-monochrome.png"
  },
  "package": "com.nexus33.verix",
  "versionCode": 1,
  "predictiveBackGestureEnabled": false
},
"web": {
  "favicon": "./assets/images/verix_logo.png",
  "bundler": "metro"
},
"plugins": [
  "expo-router",
  "expo-font",
  "expo-secure-store",
  "expo-status-bar",
  "expo-sharing",
  [
    "expo-splash-screen",
    {
      "image": "./assets/images/verix_logo.png",
      "resizeMode": "contain",
      "backgroundColor": "#0D1B2A"
    }
  ],
  "expo-asset",
  "@react-native-community/datetimepicker",
  "expo-web-browser",
  "expo-mail-composer",
```

```
[
  "expo-local-authentication",
  {
    "faceIDPermission": "Allow VeriX to use Face ID for quick, secure sign-in."
  }
],
[
  "expo-image-picker",
  {
    "photosPermission": "VeriX needs access to your photo library to attach images to support requests and evaluation records.",
    "cameraPermission": "VeriX needs access to your camera to take photos to attach to support requests and evaluation records.",
    "microphonePermission": false
  }
],
"experiments": {
  "typedRoutes": true
},
"extra": {
  "router": {},
  "eas": {
    "projectId": "b8155c89-b50f-40fd-b73b-18d9a9a59e6e"
  },
  "iosAppStoreId": ""
},
"owner": "devnexus33s-team"
}
```

? Section 2: Apple Developer Portal Setup (SOC 2 Compliant)

To comply with **SOC 2 Least Privilege Access Control**, software developers are assigned restricted roles (*Developer / Member*) without requiring full Admin privileges on the Apple Developer Account.

Step 1: Pre-Registering the App ID (Account Holder / Admin Only)

The **Account Holder** (or designated Admin) performs a one-time registration of the App Bundle Identifier:

1. Log into [Apple Developer Portal](#).
2. Navigate to **Certificates, Identifiers & Profiles → Identifiers**.
3. Click the **+** (Add) button next to **Identifiers**.
4. Select **App IDs** → Click **Continue**.
5. Select Type: **App** → Click **Continue**.
6. Enter details:
 - **Description:** VeriX App
 - **Bundle ID:** Select **Explicit** and enter `com.nexus33.verix`
7. Leave Capabilities as default (none required to be checked).
8. Click **Continue** → Click **Register**.

“ [!NOTE] Pre-registering the App ID allows developers with standard Developer roles to trigger EAS Builds without incurring HTTP 403 Forbidden permission errors.

? Section 3: Initial Credentials & EAS Linking (One-Time Execution)

The first time an iOS build is triggered for `com.nexus33.verix`, EAS CLI requires an interactive login to validate and generate the iOS Distribution Certificate on Expo's secure servers.

Interactive Build Command:

```
npx eas-cli build -p ios --profile test
```

Interactive Console Flow:

1. **Select Team:** Choose `Nexus 33 Group LLC (Z66MZ82MC2)`.
2. **Apple ID Authentication:** Enter the Apple ID email.
3. **Password / 2FA:** Enter password and 6-digit Apple Two-Factor Verification Code.

4. **Credential Setup Prompt:** Confirm ☐ Yes when prompted "Do you want EAS to manage iOS credentials?".

EAS will detect the existing `com.nexus33.verix` App ID, generate the iOS Distribution Certificate, and store credentials securely on Expo Cloud.

?? Section 4: Automated Build Execution (`build.ps1`)

Once initial credentials are created, subsequent iOS test builds can be executed automatically using the repository wrapper script:

```
.\scripts\build.ps1 -Env test -Platform ios
```

Automated Script Steps:

1. Increments `app.json` patch version (`1.0.X`).
 2. Reads current Git commit hash and timestamp.
 3. Initiates `eas build -p ios --profile test --non-interactive --json --wait`.
 4. Compiles the `.ipa` package on Expo Cloud servers.
 5. Downloads the `.ipa` artifact locally to the project root.
-

? Section 5: TestFlight Upload & Beta Testing

Once the `.ipa` build is completed (`status: FINISHED`), submit the build to TestFlight:

```
npx eas-cli submit -p ios --profile test
```

Post-Submission Steps:

1. Log into [App Store Connect](#).
2. Go to **Apps** → **VeriX** → **TestFlight** tab.
3. Once Apple finishes binary processing (~5-10 minutes), add Internal/External Testers.

4. Testers receive an email / push notification to install VeriX directly on their iPhones via the TestFlight app.

?? Troubleshooting & Security Guidelines

Issue / Error	Root Cause	Resolution
iOS simulator build (.tar.gz) generated instead of .ipa	"simulator": true in eas.json	Set "ios": { "simulator": false } under the target profile.
Apple 403 Access Forbidden	Developer account lacks Admin rights to create App ID	Follow Section 2 (Have Account Holder register com.nexus33.verix once).
Distribution Certificate not validated for non-interactive builds	Running --non-interactive before credentials exist	Run npx eas-cli build -p ios --profile test interactively once (Section 3).

Documentation prepared for Nexus 33 Group LLC — VeriX Development Team.

Revision #3
Created 2026-08-04 15:26:57 UTC by Karen Gonzalez Mogollon
Updated 2026-08-04 15:36:22 UTC by Karen Gonzalez Mogollon