Citrix MAM iOS SDKs - Compliance SDK - Developer guide

Last updated on: 2020-04-07 11:00 EST

Overview

The Compliance SDK detects the following compliance violations:

- Disabled Device Passcode (App policy: "Device Passcode")
- Jailbroken Devices (App policy: "Block jailbroken or rooted")
- Non-compliant Device Behavior (App policy: "Non-compliant device behavior")

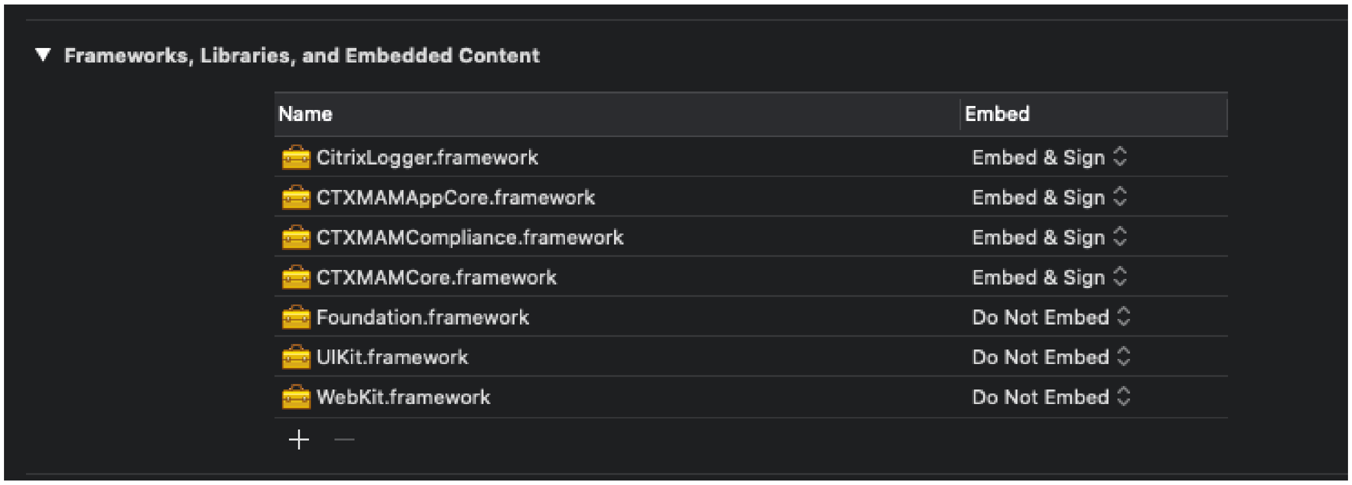
Also, it detects and enforces the following security actions:

- App Container Self Destruct (CEM Client property: "CONTAINER_SELF_DESTRUCT_PERIOD")
- User Change
- Device date and time change
- Admin App Lock and App Wipe (App policy: "Active poll period" for status poll interval)
- Wipe Data On Local App Lock (App policy: "Erase app data on lock"/WipeDataOnAppLock). Enforced on detecting Jailbroken device, no iOS Device passcode set or when the app is disabled on CEM.

Steps to integrate Compliance SDK

1. Add CTXMAMCompliance.framework to your project. If not previously added, add the CTXMAMCore.framework, CTXMAMAppCore.framework. and CitrixLogger.framework as well.

- On the project target properties under the "General" tab,
    - (In Xcode 11+), add these framework(s) in the “Frameworks, Libraries, and Embedded Content” section and select “Embed & Sign” option from the Embed dropdown.
    - (Prior to Xcode 11), add these framework(s) listed above to the "Embedded Binaries" section.

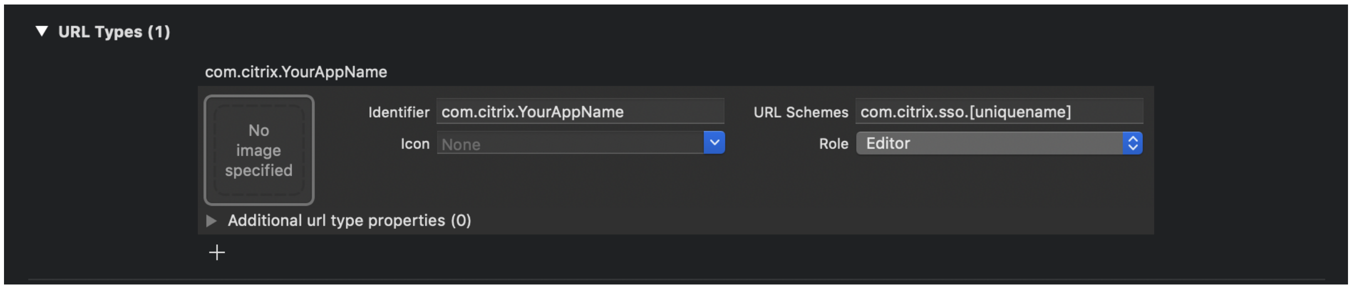


Frameworks, Libraries, and Embedded Content

Name	Embed
CitrixLogger.framework	Embed & Sign
CTXMAMAppCore.framework	Embed & Sign
CTXMAMCompliance.framework	Embed & Sign
CTXMAMCore.framework	Embed & Sign
Foundation.framework	Do Not Embed
UIKit.framework	Do Not Embed
WebKit.framework	Do Not Embed

2. On the project target properties under the "Info" tab, add a new URL Type with URL Scheme which starts with com.citrix.sso.[PACKAGEID] (e.g. com.citrix.sso.22A955C3-237B-4952-A271-567BDFFA28BA). PACKAGEID can be generated by running "uuidgen" command in your MacOS Terminal. You may skip this step is an URL Scheme with “com.citrix.sso.” prefix was previously added.

Note: Use this exact PACKAGEID in the Run Script mentioned in the subsequent step.



URL Types (1)

com.citrix.YourAppName

No image specified

Identifier: com.citrix.YourAppName

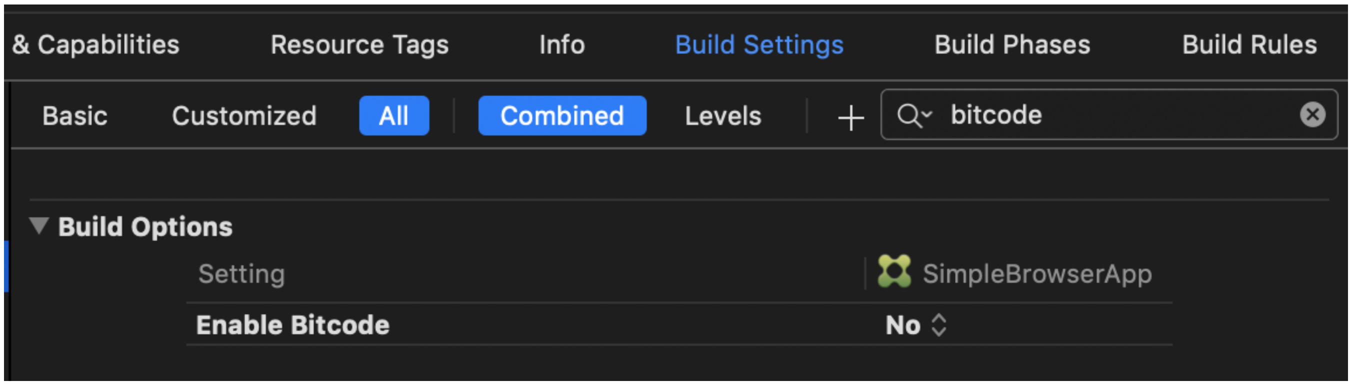
Icon: None

URL Schemes: com.citrix.sso.[uniquename]

Role: Editor

Additional url type properties (0)

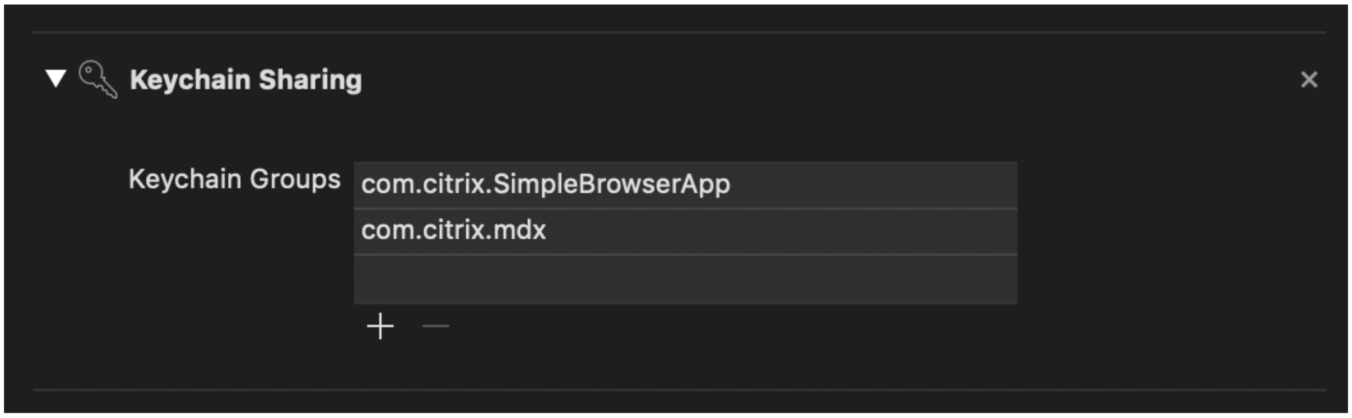
3. On the project target properties under the "Build Settings" tab, set "Enable BitCode" to "No".



Enable Bitcode No

4. On the project target properties under "Capabilities" tab, enable "Keychain Sharing".

1. Add "com.citrix.mdx" access group to the list of "Keychain Groups", if it was not previously added.



Keychain Groups
- com.citrix.SimpleBrowserApp
- com.citrix.mdx

5. Import <CTXMAMCore/CTXMAMCore.h> into your AppDelegate.m file, if it was not previously imported.

Import headers code block

```
//
// AppDelegate.m
// MyAppName
//
// Copyright © 2020 MyCompany, LLC. All rights reserved.
//

#import "AppDelegate.h"
#import <CTXMAMCore/CTXMAMCore.h>
```

6. In the AppDelegate.m in your application didFinishLaunchingWithOptions, add a call to initialize the SDKs, [CTXMAMCore initializeSDKs], if it was not previously added.

Application Did Finish Launching override

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
        // Override point for customization after application launch.

        [CTXMAMCore initializeSDKs];

        return YES;
}
```

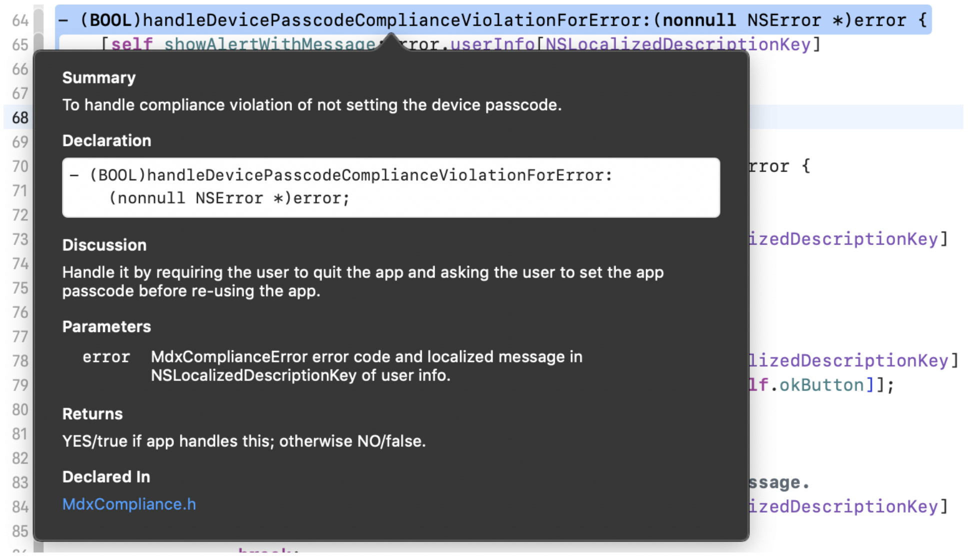
7. Integrate Compliance SDK in AppDelegate (or in another class instance)

    Compliance SDK provides a set of delegate methods to handle these compliance violations and security actions. It is recommended that the app developers handle each of these events. But if they choose not to handle some of these delegate methods, then Compliance SDK can handle those. To do so, either return NO from the implemented delegate method or do not implement the optional delegate method itself.

    - The NSError object sent in each callback belongs to error domain CTXMAMComplianceErrorDomain and has the error code of enum type CTXMAMComplianceError error codes.
    - The NSError object's userInfo has a localized reason string in key NSLocalizedDescriptionKey and could be used to present to the users.

    ```
    NSString* userInfoMsg = (NSString*) error.userInfo[NSLocalizedDescriptionKey];
    ```

    - In most cases (all but CTXMAMCompliance_Violation_EDP_WarnUser and CTXMAMCompliance_Violation_EDP_InformUser) on receiving these compliance callbacks, it is expected for the app to block the app usage.
    - More information on handling these callbacks can be found in the Quick Help of these delegate methods. For example, for the handleDevicePasscodeComplianceViolationForError: delegate method,



```
- (BOOL)handleDevicePasscodeComplianceViolationForError:(nonnull NSError *)error {
    [self showAlertWithMessage:error.userInfo[NSLocalizedDescriptionKey]
```

Summary
To handle compliance violation of not setting the device passcode.

Declaration

```
- (BOOL)handleDevicePasscodeComplianceViolationForError:
    (nonnull NSError *)error;
```

Discussion
Handle it by requiring the user to quit the app and asking the user to set the app passcode before re-using the app.

Parameters
error   MdxComplianceError error code and localized message in NSLocalizedDescriptionKey of user info.

Returns
YES/true if app handles this; otherwise NO/false.

Declared In
MdxCompliance.h

    - Currently, all the policy enforcements are triggered in the didBecomeActive event. So a notification will be fired every time the app becomes active.
    - These callbacks will asynchronously be sent to the app on the main queue.

    To integrate Compliance SDK,

    a. Import <CTXMAMCompliance/CTXMAMCompliance.h>.
    b. Conform your AppDelegate class to Compliance SDK delegate.
    c. Add all the delegate methods.

AppDelegate

```
#import "AppDelegate.h"
#import <CTXMAMCore/CTXMAMCore.h>
#import "AppDelegate+ComplianceSDKDelegate.h"
```

```
@interface AppDelegate (ComplianceSDKDelegate) <CTXMAMComplianceDelegate>
@end

@implementation AppDelegate

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {
    // Override point for customization after application launch.

    [CTXMAMCore initializeSDKs];

    [CTXMAMCompliance sharedInstance].delegate = self;

    return YES;
}


#pragma mark - Compliance SDK delegate methods
- (BOOL)handleAdminLockAppSecurityActionForError:(nonnull NSError *)error {
    [self showAlertWithMessage:error.userInfo[NSLocalizedDescriptionKey]
                    andActions:@[[self logonButtonWithTitle:@"Sign On" andErrorContext:error]]];
    return YES;
}

- (BOOL)handleAdminWipeAppSecurityActionForError:(nonnull NSError *)error {
    [self showAlertWithMessage:error.userInfo[NSLocalizedDescriptionKey]
                    andActions:@[[self logonButtonWithTitle:@"Sign On" andErrorContext:error]]];
    return YES;
}

- (BOOL)handleAppDisabledSecurityActionForError:(nonnull NSError *)error {
    [self showAlertWithMessage:error.userInfo[NSLocalizedDescriptionKey]
                    andActions:@[self.quitButton]];
    return YES;
}

- (BOOL)handleContainerSelfDestructSecurityActionForError:(nonnull NSError *)error {
    [self showAlertWithMessage:error.userInfo[NSLocalizedDescriptionKey]
                    andActions:@[[self logonButtonWithTitle:@"Continue" andErrorContext:error]]];
    return YES;
}

- (BOOL)handleDateAndTimeChangeSecurityActionForError:(nonnull NSError *)error {
    [self showAlertWithMessage:error.userInfo[NSLocalizedDescriptionKey]
                    andActions:@[[self logonButtonWithTitle:@"Sign On" andErrorContext:error]]];
    return YES;
}

- (BOOL)handleDevicePasscodeComplianceViolationForError:(nonnull NSError *)error {
    [self showAlertWithMessage:error.userInfo[NSLocalizedDescriptionKey]
                    andActions:@[self.quitButton]];
    return YES;
}

- (BOOL)handleEDPComplianceViolationForError:(nonnull NSError *)error {
        switch (error.code) {
            case CTXMAMCompliance_Violation_EDP_BlockApp:
                [self showAlertWithMessage:error.userInfo[NSLocalizedDescriptionKey]
                                andActions:@[self.quitButton]];
                break;
            case CTXMAMCompliance_Violation_EDP_WarnUser:
                 /// TODO: Need not to block app view
                 [self showAlertWithMessage:error.userInfo[NSLocalizedDescriptionKey]
                                 andActions:@[self.quitButton, self.okButton]];
                break;
            case CTXMAMCompliance_Violation_EDP_InformUser:
                  /// TODO: Need not to block app view
                  /// TODO: Show an alert of unobtrusive toast message.
                [self showAlertWithMessage:error.userInfo[NSLocalizedDescriptionKey]
                                andActions:@[self.okButton]];
                 break;
            default:
                [self showAlertWithMessage:error.userInfo[NSLocalizedDescriptionKey]
                                andActions:@[self.quitButton]];
                break;
        }
        return YES;
}

- (BOOL)handleJailbreakComplianceViolationForError:(nonnull NSError *)error {
    [self showAlertWithMessage:error.userInfo[NSLocalizedDescriptionKey]
                    andActions:@[self.quitButton]];
    return YES;
}

- (BOOL)handleUserChangeSecurityActionForError:(nonnull NSError *)error {
    [self showAlertWithMessage:error.userInfo[NSLocalizedDescriptionKey]
                    andActions:@[self.quitButton]];
    return YES;
}


#pragma mark - Helper methods
- (void) showAlertWithMessage:(NSString *)message andActions:(NSArray<UIAlertAction*>*) actions
{
// FIXME: Sample code. This code is only for documentation purposes. App developers must code their own
implementation.
//
    NSString * title = [NSString stringWithFormat:@"Compliance Error"];
    NSString * msg = [NSString stringWithFormat:@"%@", message];

    UIAlertController* alert = [UIAlertController alertControllerWithTitle:title
                                                                   message:msg
                                                            preferredStyle:UIAlertControllerStyleAlert];

    for (UIAlertAction* action in actions) {
        [alert addAction:action];
    }

    /// TODO: Present this alert on a new window or any existing window...
    dispatch_async(dispatch_get_main_queue(), ^{
        UIViewController *topController = [UIApplication sharedApplication].keyWindow.rootViewController;
        [topController presentViewController:alert animated:YES completion:nil];
    });
}


#pragma mark -
-(UIAlertAction*) quitButton
{
    UIAlertAction *action = [UIAlertAction actionWithTitle:@"Quit"
                                                      style:UIAlertActionStyleDestructive
                                                    handler:^(UIAlertAction * _Nonnull action) {
                                                        exit(0);
                                                    }];
    return action;
}
-(UIAlertAction*) okButton
{
    UIAlertAction *action = [UIAlertAction actionWithTitle:@"OK"
                                                      style:UIAlertActionStyleCancel
                                                    handler:^(UIAlertAction * _Nonnull action) {
                                                    }];
    return action;
}

-(UIAlertAction*) logonButtonWithTitle:(NSString*)title andErrorContext:(NSError*)error
{
    UIAlertAction *action = [UIAlertAction actionWithTitle:title
                                                      style:UIAlertActionStyleDefault
                                                    handler:^(UIAlertAction * _Nonnull action) {
            [[CTXMAMCompliance sharedInstance] performLogonWithErrorContext:error
```

```
                                                            completionHandler:^(BOOL success) {
                                                              NSLog(@"Logon request result:%d", success);
                                                            }];
                                                     }];
    return action;
}

@end
```

8. Back on the project target properties under the "Build Phases" tab add a new "Run Script" phase with the following script.

MDX File Creation Script

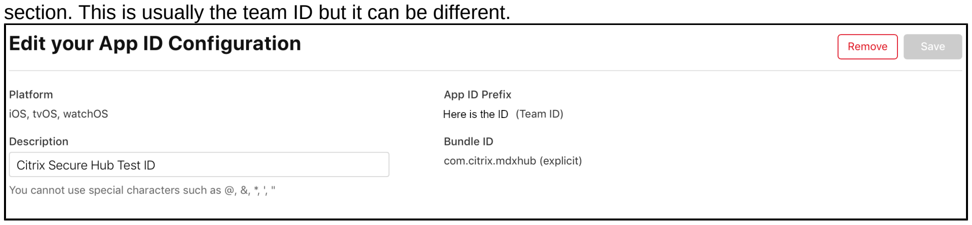
```
# Type a script or drag a script file from your workspace to insert its path.
export STOREURL="http://yourstore.yourdomain.com"
export APPTYPE="sdkapp"
export PACKAGEID="" #Please run uuidgen at the command line and paste the output value in the PACKAGEID
variable, This has be the same UUID which was generated in step 2.
export APPIDPREFIX=""#Your apps prefix id which can be found in your apple's developer account page.
export TOOLKIT_DIR="$PROJECT_DIR/../Tools"

if [ -z "${PACKAGEID}" ]
then
    echo "PACKAGEID variable was not found or was empty, please run uuidgen at the command line and paste the
output value in the PACKAGEID variable in your post build script."
    exit 1
fi

if [ -z "${APPIDPREFIX}" ]
then
    echo "APPIDPREFIX variable was not found or was empty, please refer to the \"how to\" document located in
the documentation folder of the SDK package on where to find your Apple's application prefix ID."
exit 1
fi
if [! -d $TOOLKIT_DIR/logs ]
then
    mkdir $TOOLKIT_DIR/logs
fi

"$TOOLKIT_DIR/CGAppCLPrepTool" SdkPrep -in "$CONFIGURATION_BUILD_DIR/$EXECUTABLE_FOLDER_PATH" -out
"$CONFIGURATION_BUILD_DIR/$EXECUTABLE_NAME.mdx" -storeURL "${STOREURL}" -appType "${APPTYPE}" -packageId
"${PACKAGEID}" -entitlements "$SRCROOT/$PROJECT/$PROJECT.entitlements" -appIdPrefix "${APPIDPREFIX}" -
minPlatform "9.0"
```

a. Ensure that you paste the same PACKAGEID used in the above step to add the new unique URL Scheme with the prefix "com.citrix.sso.", in the PACKAGEID field of this script.
b. Ensure that you put your app id prefix in the APPIDPREFIX field. This ID can be found in the Apple Developer portal under the AppIDs section. This is usually the team ID but it can be different.



Edit your App ID Configuration

Platform
iOS, tvOS, watchOS

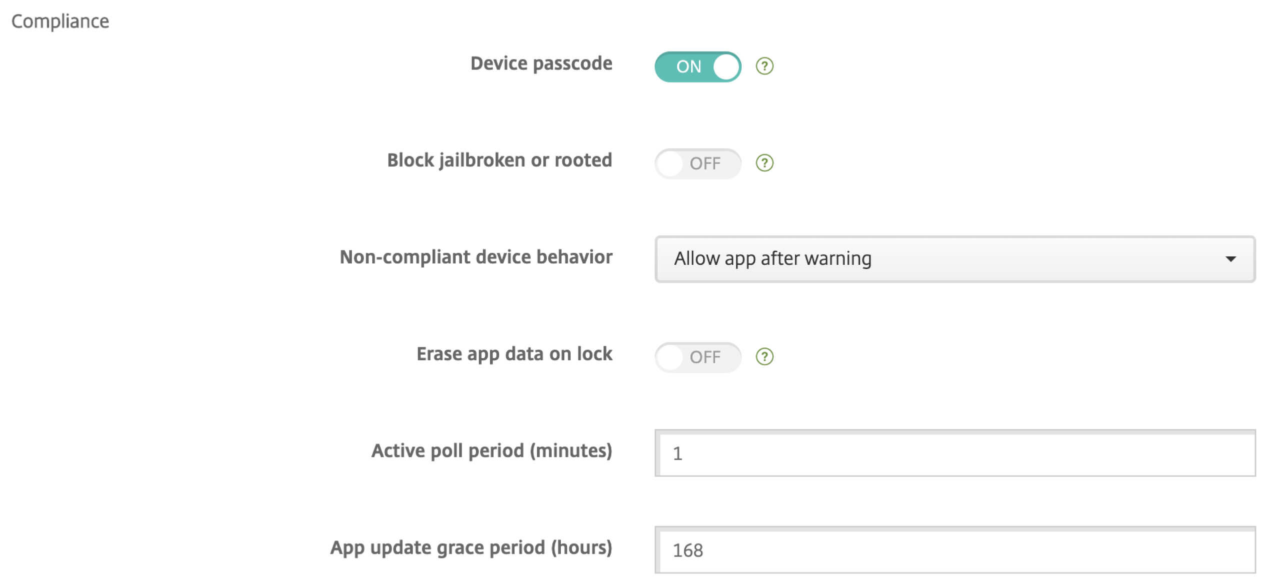
App ID Prefix
Here is the ID (Team ID)

Description
Citrix Secure Hub Test ID
You cannot use special characters such as @, &, *, ', "

Bundle ID
com.citrix.mdxhub (explicit)

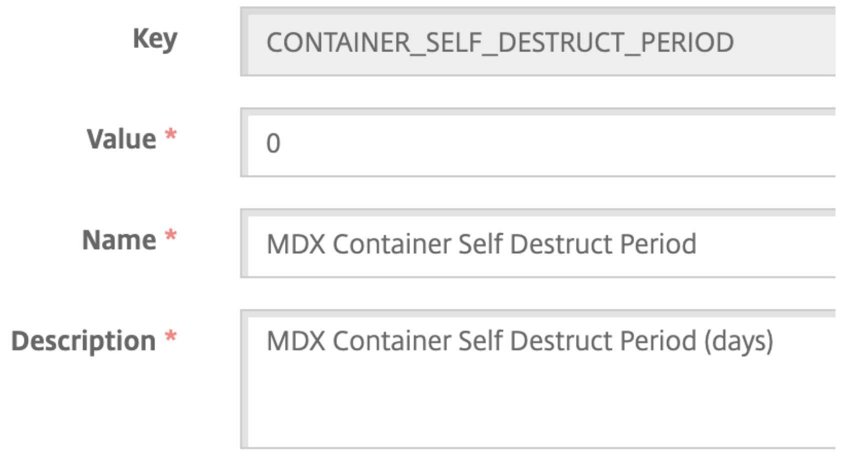
c. Ensure that the TOOLKIT_DIR property points to the Tools folder in the SDK package.

9. The above step will generate a .mdx package that can be uploaded to Citrix Endpoint Management (CEM). This package will have the following app policies for Compliance SDK.


Compliance

Device passcode	ON
Block jailbroken or rooted	OFF
Non-compliant device behavior	Allow app after warning
Erase app data on lock	OFF
Active poll period (minutes)	1
App update grace period (hours)	168

Also, the Container Self Destruct Period policy can be found or added in CEM's Client Properties.



Key	CONTAINER_SELF_DESTRUCT_PERIOD
Value *	0
Name *	MDX Container Self Destruct Period
Description *	MDX Container Self Destruct Period (days)

Embedding your IPA file into the MDX file for publication on the CEM Server

1. Create you IPA by going thru the archiving process in Xcode.
2. Run the following command:

Embedding IPA File

```
export TOOLKIT_DIR="$PROJECT_DIR/../Tools"
export IPA_FILE_PATH="Provide IPA File Path"
export EXECUTABLE_NAME="Provide the name for your application's app folder."
#CONFIGURATION_BUILD_DIR comes from Xcode.


"$TOOLKIT_DIR/CGAppCLPrepTool" SetInfo -in "$CONFIGURATION_BUILD_DIR/$EXECUTABLE_NAME.mdx" -out
"$CONFIGURATION_BUILD_DIR/$EXECUTABLE_NAME.mdx"  -embedBundle "${IPA_FILE_PATH}"
```

Sample App

A sample app called "Simple Browser app" is included in the downloads folder of the iOS MAM SDKs. All the iOS MAM SDKs are included and used in this sample app. Developers should be able to directly launch and verify the functionality of the SDKs and take similar steps for handling the delegate callbacks as shown in the app.

Extra Notes

You are required to have icons set up for your app. In particular, your 60x60 icon (the 20x20@3x icon) must be in PNG format with the name "AppIcon60x60.png"