

# COCOA 1.4.0 障害は なぜ起きたのか

ARIYAMA Keiji - デジタル庁Android（アプリ）エンジニア / 有限会社シーリス代表



## 有山 圭二 (ARIYAMA Keiji)

デジタル庁 Android (アプリ) エンジニア  
C-LIS CO., LTD. 代表

**Androidアプリ開発者**



@keiji\_ariyama



keiji

# 免責事項

## Disclaimer

本資料の内容は、

技術情報の提供を通じた技術的なコミュニケーションを目的としており、所属または関与する組織の公式見解、方針等を述べるものではありません。

# 接触確認アプリ

COVID-19 **C**ontact-**C**onfirming **A**pplication

“厚生労働省は、新型コロナウイルス感染症の拡大防止に資するよう、新型コロナウイルス感染症対策テックチームと連携して、新型コロナウイルス接触確認アプリ（COCOA ※）を開発しました。ご自身のスマートフォンにインストールして、利用いただきますようお願いいたします”

(厚生労働省ウェブサイト より)



接触確認アプリ

[https://www.mhlw.go.jp/stf/seisakunitsuite/bunya/cocoa\\_00138.html](https://www.mhlw.go.jp/stf/seisakunitsuite/bunya/cocoa_00138.html)

## Android版



<https://play.google.com/store/apps/details?id=jp.go.mhlw.covid19radar&hl=ja>

## iOS版



<https://apps.apple.com/jp/app/id1516764458>

# COCOA Open Source Project

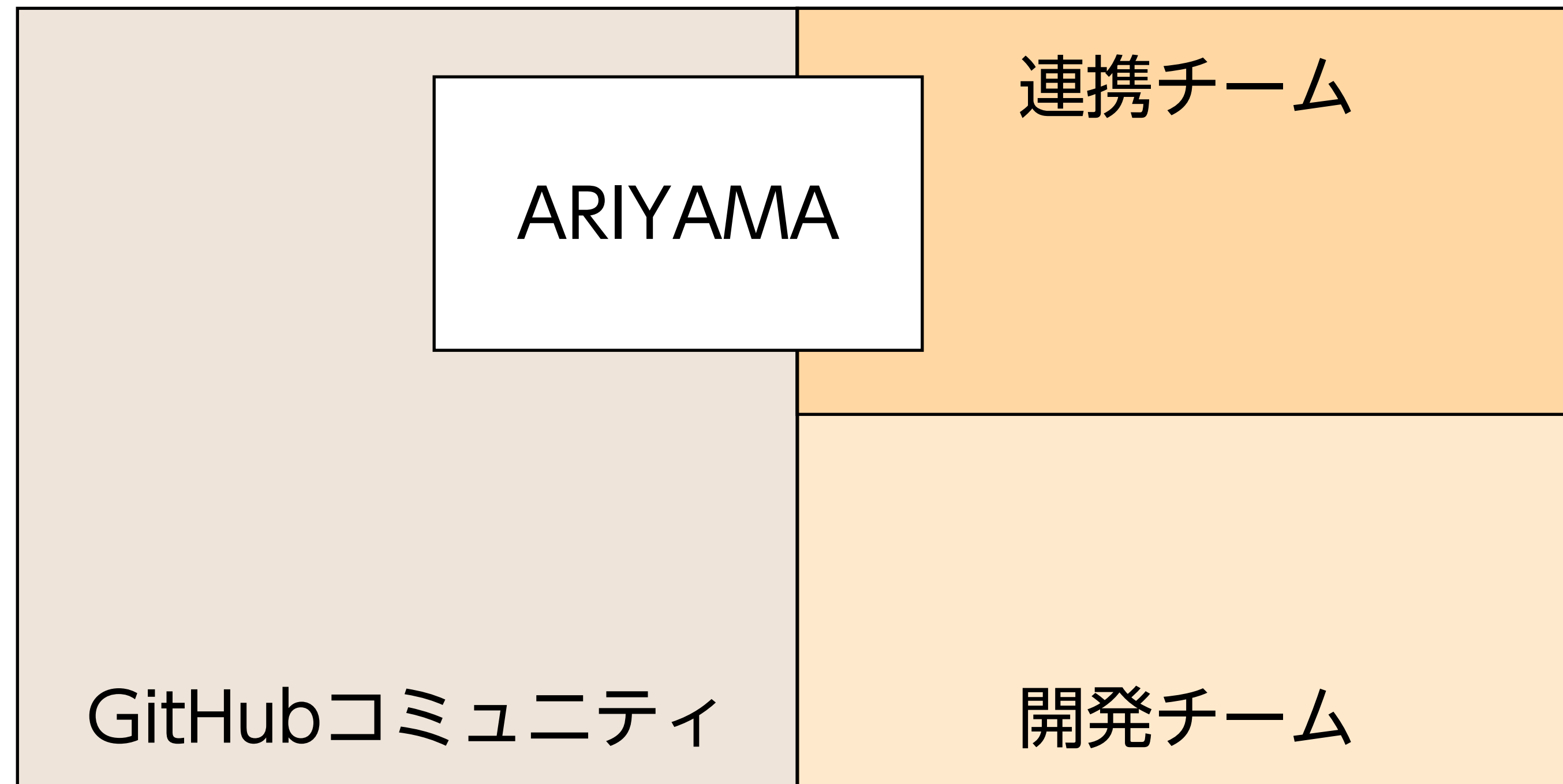
- Webサイト

<https://cocoa-mhlw.github.io/cocoa/>

- GitHub リポジトリ

<https://github.com/cocoa-mhlw/cocoa>

# COCOA開発体制 概略図



# 目次

なにが起きたのか

障害は、なぜ起きたのか

今後の対応

検討中の再発防止策



なにが起きたのか

# v1.4.0 リリース

2021/11/25 15:00

# COCOA v1.4.0 が起動しない #517

🔒 Closed

b-wind opened this issue 13 days ago · 39 comments



b-wind commented 13 days ago · edited by cocoa-dev008



## 不具合の内容 / Describe the bug

先ほど App Store / Play Store にリリースされた v1.4.0 をインストールすると起動直後にエラーで落ちている様に見えます。

## 再現手順 / Steps to reproduce

1. COCOA を起動する

## 期待される挙動 / Expected behavior

1. COCOA が正常に起動する。

<https://github.com/cocoa-mhlw/cocoa/issues/517>



keiji commented 13 days ago

Collaborator



Android版でも発生しているという情報がありますね。  
すぐに調査します。



b-wind changed the title ~~iOS 版 COCOA v1.4.0 が起動しない~~ COCOA v1.4.0 が起動しない 13 days ago



anatawa12 commented 13 days ago • edited ▾



参考になるかもわかりませんが、クラッシュレポートです[Covid19Radar.iOS-2021-11-25-162951.zip](#)



❤️ 10



keiji commented 13 days ago • edited ▾

Collaborator



@vvakame からログを提供してもらった！

17:07



59%

## Stack trace

```
android.runtime.JavaProxyThrowable:  
System.ArgumentOutOfRangeException: The added  
or subtracted value results in an un-representable  
DateTime.
```

Xamarin android App crashes with error android.runtime.JavaProxyThrowable reported in Google Play Console Vitals  
<https://docs.microsoft.com/en-us/answers/questions/418448/xamarin-android-app-crashes-with-error-androidrunt.html>

# エラー発生箇所

```
private void MigrateDateTimeToEpoch(string dateTimeKey, string epochKey, TimeSpan differential)
{
    string dateTimeStr = _preferencesService.GetValue(dateTimeKey, DateTime.UtcNow.ToString());

    DateTime dateTime;
    try
    {
        dateTime = DateTime.SpecifyKind(DateTime.Parse(dateTimeStr) + differential, DateTimeKind.Utc);
    }
    catch (FormatException exception)
    {
        _loggerService.Exception($"Parse dateTime FormatException occurred. {dateTimeStr}", exception);
        dateTime = DateTime.UtcNow;
    }
    _preferencesService.SetValue(epochKey, dateTime.ToUnixEpoch());
    _preferencesService.RemoveValue(dateTimeKey);
}
```

[https://github.com/cocoa-mhlw/cocoa/blob/release\\_1.4.0/Covid19Radar/Covid19Radar/Services/Migration/Migrator\\_1.3.0.cs](https://github.com/cocoa-mhlw/cocoa/blob/release_1.4.0/Covid19Radar/Covid19Radar/Services/Migration/Migrator_1.3.0.cs)



jmatsu commented 13 days ago • edited ▾



@keiji

↓で詰め込んでる `new DateTime()` だと `DateTime.MinValue` になり、`MinValue/MaxValue` に加算減算が定義されていない、という仕様のようなので。現在時刻が欲しいなら `DateTime.Now` にならないといけなさそうです。

[cocoa/Covid19Radar/Covid19Radar/Services/Migration/Migrator\\_1\\_2\\_2.cs](#)

Line 169 in 60e00c7

```
169     _preferencesService.SetValue(preferenceKey, new DateTime().ToString());
```

<https://ideone.com/XJiYnM>

▶ Android11, Pixel3 XL, en\_US, JST +9 の LogCat







keiji commented 13 days ago

Collaborator



`DateTime()` は `0001/01/01 0:00:00` で、そこからさらに9時間を引こうとしているので怒られている。

`DateTime.MinValue`と同値であれば引かないようにとか考えたけど、`DateTime.MinValue`と同値ではない `0001/01/01 0:08:00` から9時間引こうとするとやっぱり落ちるよねと。

先行きは見えてきたので、修正方法を決めます。



6

# Hotfix

```
private void MigrateDateTimeToEpoch(string dateTimeKey, string epochKey, TimeSpan differential, DateTime fallbackDateTime)
{
    string dateTimeStr = _preferencesService.GetValue(dateTimeKey, fallbackDateTime.ToString());


    DateTime dateTime;
    try
    {
        dateTime = DateTime.SpecifyKind(DateTime.Parse(dateTimeStr), DateTimeKind.Utc);
    }
    catch (FormatException exception)
    {
        _loggerService.Exception($"Parse dateTime FormatException occurred. {dateTimeStr}", exception);
        dateTime = fallbackDateTime;
    }


    try
    {
        dateTime += differential;
    }
    catch (ArgumentOutOfRangeException exception)
    {
        _loggerService.Exception($"{dateTimeStr} {differential} The added or subtracted value results in an un-representable DateTime.", exception);
    }



    _preferencesService.SetValue(epochKey, dateTime.ToUnixEpoch());
    _preferencesService.RemoveValue(dateTimeKey);
}
```

Hot fix #517



<https://github.com/cocoa-mhlw/cocoa/pull/518>




 **cocoa-dev004** approved these changes 13 days ago [View changes](#)

**cocoa-dev004** left a comment  

早急な修正ありがとうございます

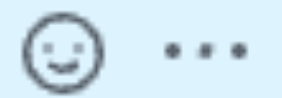
  1

 **cocoa-dev004** merged commit **fa055dd** into `release_1_4_1` 13 days ago [View details](#) [Revert](#)  
7 checks passed



keiji commented 13 days ago

Collaborator




ホットフィックス版（v1.4.1）のテストを終え、App Store, Google Play Storeへの提出を完了しました。



❤️ 12

2021/11/25 19:19

 **cocoa-dev004** merged commit **fa055dd** into **release\_1\_4\_1** 13 days ago  
7 checks passed


[View details](#) [Revert](#)

- \* 20:12 - 開発チーム 利用者からの問い合わせ状況を共有
- \* 20:13 - 開発チーム v1.1.5以前にアプリの利用を開始すると利用規約の同意日付がDateTime最小値のままとなることが原因と判明
  - \* 開発チーム 障害の再現方法を確定
- \* 20:39 - 開発チーム リリーステストの作成開始
- \* 22:40 - 開発チーム リリーステストの作成完了
- \* 23:16 - 開発チーム リリーステストの実施完了
  - \* 開発チーム 準備したv1.4.0がクラッシュする端末のアプリを製品版アプリv1.4.1へアップデート、正常起動し問題が解消していることを確認

26/11/2021 00:28



**keiji** commented 13 days ago

Collaborator  ...

ホットフィックス版 (v1.4.1) のテストを終え、App Store, Google Play Storeへの提出を完了しました。



 12

# v1.4.1 Hotfixリリース

## \* 11/26

- \* 00:28 - 開発チーム v1.4.1 審査提出完了 (Google Play, AppStore)
- \* 10:10 - 開発チーム v1.4.1 AppStoreでリリース
  - \* 開発チーム iOS端末7台で公開されたアプリv1.4.1が起動できることを確認
- \* 10:47 - GitHub iOS版の復旧報告

## \* 11/27

- \* 11:28 - 開発チーム v1.4.1 Google Playでリリース開始
- \* 11:40 - GitHub Android版の復旧報告
  - \* 開発チーム Android端末1台で公開されたアプリv1.4.1が起動できることを確認

**GitHubコミュニティのみなさん**

**ご協力、本当にありがとうございました。**

障害は、なぜ起きたのか

# この処理はなに？

## v1.3.0へのマイグレーション処理

```
private void MigrateDateTimeToEpoch(string dateTimeKey, string epochKey, TimeSpan differential)
{
    string dateTimeStr = _preferencesService.GetValue(dateTimeKey, DateTime.UtcNow.ToString());

    DateTime dateTime;
    try
    {
        dateTime = DateTime.SpecifyKind(DateTime.Parse(dateTimeStr) + differential, DateTimeKind.Utc);
    }
    catch (FormatException exception)
    {
        _loggerService.Exception($"Parse dateTime FormatException occurred. {dateTimeStr}", exception);
        dateTime = DateTime.UtcNow;
    }
    _preferencesService.SetValue(epochKey, dateTime.ToUnixEpoch());
    _preferencesService.RemoveValue(dateTimeKey);
}
```

[https://github.com/cocoa-mhlw/cocoa/blob/release\\_1.4.0/Covid19Radar/Covid19Radar/Services/Migration/Migrator\\_1.3.0.cs](https://github.com/cocoa-mhlw/cocoa/blob/release_1.4.0/Covid19Radar/Covid19Radar/Services/Migration/Migrator_1.3.0.cs)



# マイグレーション (Migration) 処理

## バージョン間で異なるデータ形式を移行する

- データ形式の異なる時期
  - v1.0.0 - v1.2.1
  - v1.2.2 - v1.2.6

# v1.0.0-v1.2.1

- アプリのデータはApplication.Propertiesに保存
  - Xamarin.Formsが用意している永続化方式
  - 実体ファイルはBinary XML形式

# Application.Propertiesの課題

- Application.Propertiesはスレッドセーフではない
- 複数スレッドからアクセスするとデータが壊れることがある
- COCOA 突然初期化される問題

<https://github.com/cocoa-mhlw/cocoa/issues/16>

# v1.2.2-v1.2.5

- アプリのデータはプラットフォーム固有の形式に保存
  - Androidでは`SharedPreferences`
  - iOSでは`NSUserDefaults`
- Application.Propertiesからの移行（マイグレーション）が追加

# v1.2.2-v1.2.5の課題

## 複数クラスにマイグレーション処理が分散していた

- 利用規約とプライバシーポリシーの合意日時

[https://github.com/cocoa-mhlw/cocoa/blob/release\\_1\\_2\\_5/Covid19Radar/Covid19Radar/Services/TermsUpdateService.cs](https://github.com/cocoa-mhlw/cocoa/blob/release_1_2_5/Covid19Radar/Covid19Radar/Services/TermsUpdateService.cs)

- 利用開始日・利用規約とプライバシーポリシーの合意フラグ

[https://github.com/cocoa-mhlw/cocoa/blob/release\\_1\\_2\\_5/Covid19Radar/Covid19Radar/Services/UserDataService.cs](https://github.com/cocoa-mhlw/cocoa/blob/release_1_2_5/Covid19Radar/Covid19Radar/Services/UserDataService.cs)

- 接触記録

[https://github.com/cocoa-mhlw/cocoa/blob/release\\_1\\_2\\_5/Covid19Radar/Covid19Radar/Services/ExposureNotificationService.cs](https://github.com/cocoa-mhlw/cocoa/blob/release_1_2_5/Covid19Radar/Covid19Radar/Services/ExposureNotificationService.cs)

# v1.2.2-v1.2.5の課題

プラットフォーム固有の移行処理をサードパーティライブラリを改変する形で記述していた

```
namespace Xamarin.ExposureNotifications
{
    public static partial class ExposureNotification
    {
        private static readonly string[] OldWorkNames = { "exposurenotification" }; // Array of old work-name.
        private static readonly string CurrentWorkName = "cocoaexposurenotification"; // Current work-name. (changed policy from `replace` to `keep`)

        // Schedule background work (Customization by COCOA)
        static Task PlatformScheduleFetch()
        {
            CancelOldWork();

            var workRequest = CreatePeriodicWorkRequest();
            EnqueueUniquePeriodicWork(workRequest);

            return Task.CompletedTask;
        }

        private static void CancelOldWork()
        {
            var workManager = WorkManager.GetInstance(Essentials.Platform.AppContext);
            foreach (var oldWorkName in OldWorkNames)
            {
                workManager.CancelUniqueWork(oldWorkName);
            }
        }
    }
}
```

[https://github.com/cocoa-mhlw/cocoa/blob/release\\_1\\_2\\_5/Covid19Radar/Xamarin.ExposureNotification/ExposureNotification.customize.android.cs](https://github.com/cocoa-mhlw/cocoa/blob/release_1_2_5/Covid19Radar/Xamarin.ExposureNotification/ExposureNotification.customize.android.cs)

# v1.2.6 マイグレーションの仕組みを導入

## 分散させず、一カ所で処理する

- fromVersion → toVersion で必要なマイグレーションを順次実行
  - fromVersion: 設定に記録される（または推測する）アプリのバージョン
  - toVersion: 現在のアプリのバージョン

# v1.2.6 リリリース

2021/09/08



# この処理はなに？

## **\*\*v1.3.0\*\*** へのマイグレーション処理

```
private void MigrateDateTimeToEpoch(string dateTimeKey, string epochKey, TimeSpan differential)
{
    string dateTimeStr = _preferencesService.GetValue(dateTimeKey, DateTime.UtcNow.ToString());

    DateTime dateTime;
    try
    {
        dateTime = DateTime.SpecifyKind(DateTime.Parse(dateTimeStr) + differential, DateTimeKind.Utc);
    }
    catch (FormatException exception)
    {
        _loggerService.Exception($"Parse dateTime FormatException occurred. {dateTimeStr}", exception);
        dateTime = DateTime.UtcNow;
    }
    _preferencesService.SetValue(epochKey, dateTime.ToUnixEpoch());
    _preferencesService.RemoveValue(dateTimeKey);
}
```

[https://github.com/cocoa-mhlw/cocoa/blob/release\\_1.4.0/Covid19Radar/Covid19Radar/Services/Migration/Migrator\\_1.3.0.cs](https://github.com/cocoa-mhlw/cocoa/blob/release_1.4.0/Covid19Radar/Covid19Radar/Services/Migration/Migrator_1.3.0.cs)

# v1.2.6 までの課題

## 日時情報の保存形式

- v1.2.6時点で、COCOAには4つの日時情報を保存
  - 使用開始日時 (StartDateTime)
  - 利用規約に合意した日時 (TermsOfServiceLastUpdateDateTime)
  - プライバシーポリシーに合意した日時 (PrivacyPolicyLastUpdateDateTime)
  - 接触確認した診断キーの最新の作成日 (LastProcessTekTimestamp)

[https://cocoa-mhlw.github.io/cocoa/docs/appendix/preference\\_specification/#v122---v126](https://cocoa-mhlw.github.io/cocoa/docs/appendix/preference_specification/#v122---v126)

# 文字列として保存された日時情報

- LastProcessTekTimestampを除いた3つの日時情報を文字列として保存
  - 使用開始日時 (StartDateTime)
  - 利用規約に合意した日時 (TermsOfServiceLastUpdateDateTime)
  - プライバシーポリシーに合意した日時 (PrivacyPolicyLastUpdateDateTime)
- 接触確認した診断キーの最新の作成日 (LastProcessTekTimestamp)

# 日時文字列はフォーマット指定なしのDateTime.ToString()の結果

## DateTime.ToString()の結果は実行環境（言語・ロケール）に依存する

```
using System;
using System.Globalization;

public class Program
{
    public static void Main()
    {
        CultureInfo culture = CultureInfo.CreateSpecificCulture("ja-JP");
        CultureInfo.CurrentCulture = culture;

        var datetime = new DateTime(2021, 12, 11);

        Console.WriteLine(datetime.ToString());
    }
}
```

2021/12/11 0:00:00

# 各ロケールでのDateTime.ToString()の実行結果

2021年12月11日 new DateTime(2021, 12, 11).ToString()

- 日本語-日本 (ja-JP)  
→ 2021/12/11 0:00:00
- 英語-アメリカ合衆国 (en-US)  
→ 12/11/2021 12:00:00 AM
- 英語-英国 (en-GB)  
→ 11/12/2021 00:00:00
- 英語-日本 (en-JP)  
→ 12/11/2021 12:00:00 AM

「英語・英国 (en-GB)」に設定している人が、  
日本に到着してすぐにCOCOAをインストール。  
利用開始後、設定を「英語・日本 (en-JP)」に切り替えたなら？

# en-GBで保存される日時文字列

"11/12/2021 00:00:00"

# en-JPで解釈される日時

```
using System;
using System.Globalization;

public class Program
{
    public static void Main()
    {
        CultureInfo culture = CultureInfo.CreateSpecificCulture("en-JP");
        CultureInfo.CurrentCulture = culture;

        var datetime = DateTime.Parse("11/12/2021 00:00:00");

        Console.WriteLine(datetime.ToString());
    }
}
```

"11/12/2021 12:00:00 AM"



# 日時情報として解釈できないケース

```
using System;
using System.Globalization;

public class Program
{
    public static void Main()
    {
        CultureInfo culture = CultureInfo.CreateSpecificCulture("en-JP");
        CultureInfo.CurrentCulture = culture;

        var datetime = DateTime.Parse("25/12/2021 00:00:00");

        Console.WriteLine(datetime.ToString());
    }
}
```

```
Unhandled exception. System.FormatException: String '25/12/2021 00:00:00' was not recognized as a valid DateTime.
  at System.DateTime.Parse(String s)
  at Program.Main()
```

# 関連する課題

- [Android] 端末の言語を変更後「プライバシーポリシーの改定」が再表示される

<https://github.com/cocoa-mhlw/cocoa/issues/49>

- 各種日付がロケール依存のフォーマットで保存される

<https://github.com/cocoa-mhlw/cocoa/issues/51>

- [iOS] 利用開始後、端末ロケールを変更すると使用開始日がリセットされ続ける

<https://github.com/cocoa-mhlw/cocoa/issues/124>

# v1.3.0のマイグレーション処理

## 文字列の日時情報をUNIX Epochへ変換する

```
public Task ExecuteAsync()
{
    if (_preferencesService.ContainsKey(START_DATETIME))
    {
        MigrateDateTimeToEpoch(START_DATETIME, PreferenceKey.StartDateTimeEpoch, TimeSpan.Zero);
    }

    if (_preferencesService.ContainsKey(TERMS_OF_SERVICE_LAST_UPDATE_DATETIME))
    {
        MigrateDateTimeToEpoch(
            TERMS_OF_SERVICE_LAST_UPDATE_DATETIME,
            PreferenceKey.TermsOfServiceLastUpdateDateTimeEpoch,
            -TIME_DIFFERENTIAL_JST_UTC
        );
    }

    if (_preferencesService.ContainsKey(PRIVACY_POLICY_LAST_UPDATE_DATETIME))
    {
        MigrateDateTimeToEpoch(
            PRIVACY_POLICY_LAST_UPDATE_DATETIME,
            PreferenceKey.PrivacyPolicyLastUpdateDateTimeEpoch,
            -TIME_DIFFERENTIAL_JST_UTC
        );
    }

    return Task.CompletedTask;
}
```

[https://github.com/cocoa-mhlw/cocoa/blob/release\\_1\\_4\\_0/Covid19Radar/Covid19Radar/Services/Migration/Migrator\\_1\\_3\\_0.cs](https://github.com/cocoa-mhlw/cocoa/blob/release_1_4_0/Covid19Radar/Covid19Radar/Services/Migration/Migrator_1_3_0.cs)

# この処理はなに？

## 文字列の日時情報をUNIX Epochへ変換する

```
private void MigrateDateTimeToEpoch(string dateTimeKey, string epochKey, TimeSpan differential)
{
    string dateTimeStr = _preferencesService.GetValue(dateTimeKey, DateTime.UtcNow.ToString());

    DateTime dateTime;
    try
    {
        dateTime = DateTime.SpecifyKind(DateTime.Parse(dateTimeStr) + differential, DateTimeKind.Utc);
    }
    catch (FormatException exception)
    {
        _loggerService.Exception($"Parse dateTime FormatException occurred. {dateTimeStr}", exception);
        dateTime = DateTime.UtcNow;
    }
    _preferencesService.SetValue(epochKey, dateTime.ToUnixEpoch());
    _preferencesService.RemoveValue(dateTimeKey);
}
```

[https://github.com/cocoa-mhlw/cocoa/blob/release\\_1.4.0/Covid19Radar/Covid19Radar/Services/Migration/Migrator\\_1.3.0.cs](https://github.com/cocoa-mhlw/cocoa/blob/release_1.4.0/Covid19Radar/Covid19Radar/Services/Migration/Migrator_1.3.0.cs)

障害は、なぜ起きたのか

# なぜこのコードを書いたのか

```
private void MigrateDateTimeToEpoch(string dateTimeKey, string epochKey, TimeSpan differential)
{
    string dateTimeStr = _preferencesService.GetValue(dateTimeKey, DateTime.UtcNow.ToString());

    DateTime dateTime;
    try
    {
        dateTime = DateTime.SpecifyKind(DateTime.Parse(dateTimeStr) + differential, DateTimeKind.Utc);
    }
    catch (FormatException exception)
    {
        _loggerService.Exception($"Parse dateTime FormatException occurred. {dateTimeStr}", exception);
        dateTime = DateTime.UtcNow;
    }
    _preferencesService.SetValue(epochKey, dateTime.ToUnixEpoch());
    _preferencesService.RemoveValue(dateTimeKey);
}
```

[https://github.com/cocoa-mhlw/cocoa/blob/release\\_1.4.0/Covid19Radar/Covid19Radar/Services/Migration/Migrator\\_1.3.0.cs](https://github.com/cocoa-mhlw/cocoa/blob/release_1.4.0/Covid19Radar/Covid19Radar/Services/Migration/Migrator_1.3.0.cs)

# 時差補正の要件

- 日時情報 (DateTime) の文字列にはタイムゾーンが含まれていない。  
しかし、利用規約・プライバシーポリシーの同意日時についてはJSTとして取り扱う必要がある
  - 使用開始日時 (StartDateTime) - UTC
  - 利用規約に合意した日時 (TermsOfServiceLastUpdateDateTime) - JST
  - プライバシーポリシーに合意した日時 (PrivacyPolicyLastUpdateDateTime) - JST
- 開発チームより指摘を受けて対応

<https://github.com/cocoa-mhlw/cocoa/commit/2b7b19de75b42e44cb1bb1c87406c99af9bd7754#diff-180bad4c321871e2b4bf1cd58ce78d3040ccd6a64ec0937fd84e0b3514a00cc5>

なぜタイムゾーンの補正に  
加減算を用いたのか



# 入力文字列に対する認識の甘さ

- DateTimeの計算で（下限値を下回る・上限値を上回る）ときに例外が発生するという知識が不足していた
- 入力される値に下限値（0001/01/01 00:00:00）が存在するという認識が不足していた

# 検討した事項

## `DateTime.ToUniversalTime()`

- `DateTime.ToUniversalTime()`は、端末のロケールに設定されている地域のタイムゾーンを「ローカルタイム (Local)」として、時差の補正を行う
- 今回の場合、`DateTime`の元になる文字列にタイムゾーンの情報はない。その上でタイムゾーンを固定 (JSTまたはUTC) して時差を補正する必要がある
- 端末に設定されているロケールが「日本 (JST)」である保証はない

# 検討した事項

## TimeZoneInfo.ConvertTimeToUtc

- ConvertTimeToUtcを実行する上で必要な「タイムゾーン (TimeZoneInfo)」を取得する際に用いるTimeZoneInfo.FindSystemTimeZoneByIdに指定する値が、プラットフォームによって異なる。
  - Windowsでは "Tokyo Standard Time" ← Microsoft公式ドキュメントで紹介
  - Linuxでは "Asia/Tokyo" ← Xamarinは (?) こちら

**プラットフォームごとの挙動の違いが、利用を忌避する要因になった**

# 入力文字列に対する認識の甘さ

- DateTimeの計算で（下限値を下回る・上限値を上回る）ときに例外が発生するという知識が不足していた
- 入力される値の範囲に下限値（0001/01/01 00:00:00）が存在するという認識が不足していた

# 一度は合意しないとアプリが利用できない

- 使用開始日時 (StartDateTime) - UTC
- 利用規約に合意した日時 (TermsOfServiceLastUpdateDateTime) - JST
- プライバシーポリシーに合意した日時  
(PrivacyPolicyLastUpdateDateTime) - JST

なぜ、利用規約合意日が

**0001/01/01 00:00:00**

になるのか

# v1.2.2のマイグレーション

```
private async Task MigrateTermAsync(TermsType termsType, bool isAgree)
{
    var applicationPropertyKey = termsType == TermsType.TermsOfService ?
        APPLICATION_PROPERTY_TERMS_OF_SERVICE_LAST_UPDATE_DATE_KEY : APPLICATION_PROPERTY_PRIVACY_POLICY_LAST_UPDATE_DATE_KEY;
    var preferenceKey = termsType == TermsType.TermsOfService ?
        PREFERENCE_KEY_TERMS_OF_SERVICE_LAST_UPDATE_DATETIME : PREFERENCE_KEY_PRIVACY_POLICY_LAST_UPDATE_DATETIME;

    if (_preferencesService.ContainsKey(applicationPropertyKey))
    {
        return;
    }

    if (isAgree)
    {
        if (_applicationPropertyService.ContainsKey(applicationPropertyKey))
        {
            var lastUpdateDate = _applicationPropertyService.GetProperties(applicationPropertyKey).ToString();
            _preferencesService.SetValue(preferenceKey, lastUpdateDate);
        }
        else
        {
            _preferencesService.SetValue(preferenceKey, new DateTime().ToString());
        }
    }

    await _applicationPropertyService.Remove(applicationPropertyKey);
}
```

[https://github.com/cocoa-mhlw/cocoa/blob/release\\_1\\_4\\_0/Covid19Radar/Covid19Radar/Services/Migration/Migrator\\_1\\_2\\_2.cs](https://github.com/cocoa-mhlw/cocoa/blob/release_1_4_0/Covid19Radar/Covid19Radar/Services/Migration/Migrator_1_2_2.cs)

[https://github.com/cocoa-mhlw/cocoa/blob/release\\_1\\_2\\_5/Covid19Radar/Covid19Radar/Services/TermsUpdateService.cs](https://github.com/cocoa-mhlw/cocoa/blob/release_1_2_5/Covid19Radar/Covid19Radar/Services/TermsUpdateService.cs)

# 把握していなかった変更

- 利用規約、プライバシーポリシーの合意日時を記録するようになったのは v1.2.0（2020/12/03リリース）から
  - プライバシーポリシー、利用規約の改訂があった
- v1.1.5以前は合意したかどうか（IsOptined, IsPolicyAccepted）のみ管理

接触確認アプリ「COCOA」の修正版（「1.2.0」）の配布を開始しました。

[https://www.mhlw.go.jp/stf/newpage\\_15244.html](https://www.mhlw.go.jp/stf/newpage_15244.html)



# v1.2.2のマイグレーション

## v1.1.5以前からのマイグレーションで発生

```
private async Task MigrateTermAsync(TermsType termsType, bool isAgree)
{
    var applicationPropertyKey = termsType == TermsType.TermsOfService ?
        APPLICATION_PROPERTY_TERMS_OF_SERVICE_LAST_UPDATE_DATE_KEY : APPLICATION_PROPERTY_PRIVACY_POLICY_LAST_UPDATE_DATE_KEY;
    var preferenceKey = termsType == TermsType.TermsOfService ?
        PREFERENCE_KEY_TERMS_OF_SERVICE_LAST_UPDATE_DATETIME : PREFERENCE_KEY_PRIVACY_POLICY_LAST_UPDATE_DATETIME;

    if (_preferencesService.ContainsKey(applicationPropertyKey))
    {
        return;
    }

    if (isAgree)
    {
        if (_applicationPropertyService.ContainsKey(applicationPropertyKey))
        {
            var lastUpdateDate = _applicationPropertyService.GetProperties(applicationPropertyKey).ToString();
            _preferencesService.SetValue(preferenceKey, lastUpdateDate);
        }
        else
        {
            _preferencesService.SetValue(preferenceKey, new DateTime().ToString());
        }
    }

    await _applicationPropertyService.Remove(applicationPropertyKey);
}
```

[https://github.com/cocoa-mhlw/cocoa/blob/release\\_1\\_4\\_0/Covid19Radar/Covid19Radar/Services/Migration/Migrator\\_1\\_2\\_2.cs](https://github.com/cocoa-mhlw/cocoa/blob/release_1_4_0/Covid19Radar/Covid19Radar/Services/Migration/Migrator_1_2_2.cs)

[https://github.com/cocoa-mhlw/cocoa/blob/release\\_1\\_2\\_5/Covid19Radar/Covid19Radar/Services/TermsUpdateService.cs](https://github.com/cocoa-mhlw/cocoa/blob/release_1_2_5/Covid19Radar/Covid19Radar/Services/TermsUpdateService.cs)

なぜ

0001/01/01 00:00:00 が  
いまだに残っているのか？

v1.2.0以降、プライバシーポリシーと  
利用規約が改訂されているのだから、  
その時点の日付になっているはずでは？

# 利用規約の再同意プロセスが使われた（表示された） ことがない

```
{
  "privacy_policy" : {
    "text" : "We will retain in the management system the processing numbers required to register people who have tested positive for COVID-19 in COCOA to respond to inquiries. We have revised the Privacy Policy due to this. Please confirm the revised content from the link below.",
    "update_date" : "2021/12/01 13:00:00"
  }
}
```

```
{
  "terms_of_service" : {
    "text": "",
    "update_date" : ""
  },
  "privacy_policy" : {
    "text" : "We will retain in the management system the processing numbers required to register people who have tested positive for COVID-19 in COCOA to respond to inquiries. We have revised the Privacy Policy due to this. Please confirm the revised content from the link below.",
    "update_date" : "2021/12/01 13:00:00"
  }
}
```

**v1.1.5で利用規約に合意していた場合、合意日時は更新されない（0001/01/01 00:00:00のまま）**

障害は、なぜ起きたのか  
まとめ

# 課題に対する認識の甘さ・実力不足

- タイムゾーン補正（JSTからUTC）方法の不適切な選択
- 言語（C#）仕様への理解が不十分なまま実装をした
- 過去バージョンの仕様の調査が不足していた

アクシデントレポート「v1.4.0リリースで発生した障害について」 #544

<https://github.com/cocoa-mhlw/cocoa/pull/544>

# COCOA 1.4.0 障害は なぜ起きたのか

誰かのせいにしたいが  
自分の顔しか思い浮かばない

ARIYAMA Keiji - デジタル庁Android（アプリ）エンジニア / 有限会社シーリス代表

# 今後の対応

- マイグレーション処理を改善する

<https://github.com/cocoa-mhlw/cocoa/issues/520>

- アクシデントレポートを書く

<https://github.com/cocoa-mhlw/cocoa/pull/544>

- 再発防止策の検討・実施



# 検討中の再発防止策

## 障害発生時の影響を低減する方策も含む

- 日時情報の取り扱いについて定める
  - 内部ではUNIX Epochとして保存する。また、文字列表現をRFC 3339に統一することでロケールの違いによる挙動の違いを吸収する
- ユニットテストを整備する。特にコードカバレッジを計測することで、ユニットテストがカバーできていないところを可視化する
- ローリングアップデート（段階的な公開）を実施することで、リリースを中止できるようにする

# 検討中の再発防止策

## 障害発生時の影響を低減する方策も含む

- 正式リリース前にBeta版を配布することで、より多くの利用者、端末でテストできるようにする
- リリース予定日・リリース状況（ローリングアップデート含む）を、あらかじめGitHubなどで公表し、公開時の目の数を増やすことで障害の発生をいち早く検出する
- リリース時期（ローリングアップデート時含む）には、SNSの観測をオペレーションとして行う

# 検討していない再発防止策

- ~~コードの取り込みに当たっては必ずダブルチェック、可能であればトリプルチェックを行う~~

ご清聴ありがとうございました