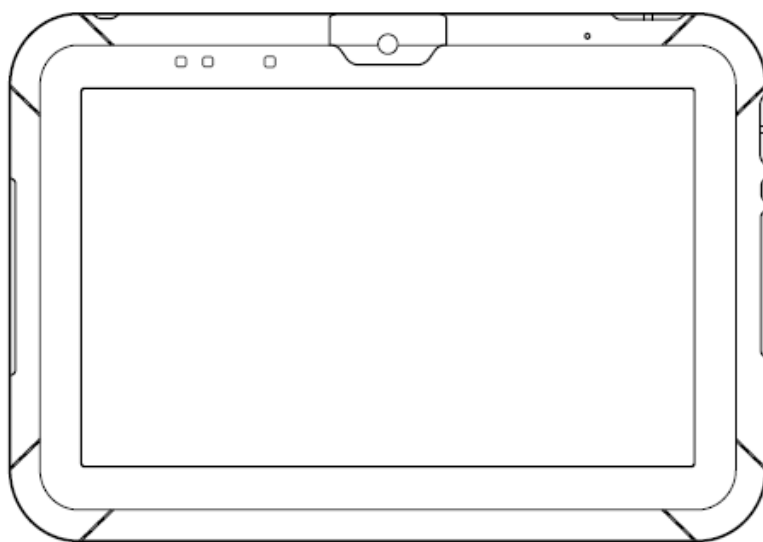


## 認証ライブラリ マニュアル

このマニュアルは、認証ライブラリの仕様について記載します。



Google™、Googleロゴ、Android™、AndroidロゴはGoogle, Inc.の商標または登録商標です。

その他本文中に記載されている会社名および商品名は、各社の商標または登録商標です。

#### ご注意

- このソフトウェアおよびマニュアルの、一部または全部を無断で使用、複製することはできません。
- このソフトウェアおよびマニュアルは、本製品の使用許諾契約書のもとでのみ使用することができます。
- このソフトウェアおよびマニュアルを運用した結果の影響については、一切の責任を負いかねますのでご了承ください。
- このソフトウェアの仕様、およびマニュアルに記載されている事柄は、将来予告なしに変更することがあります。
- このマニュアルの著作権はカシオ計算機株式会社に帰属します。
- 本書中に含まれている画面表示は、実際の画面とは若干異なる場合があります。予めご了承ください。

© 2014 カシオ計算機株式会社

Microsoft, MS, ActiveSync, Active Desktop, Outlook, Windows, Windows NT, および Windows ロゴは、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。Microsoft 社の製品は、OEM 各社に、Microsoft Corporation の 100%出資子会社である Microsoft Licensing, Inc.によりライセンス供与されています。

# 目次

1.	概要	1
1.1	概要	1
1.2	動作環境	1
1.2.1	対象機種	1
1.2.2	対象OS	1
1.2.3	プログラミング言語	1
1.2.4	必要なシステム	1
1.2.5	必要なソフトウェア	2
2.	ライブラリ	3
2.1	Package名称	3
2.2	関数一覧	3
2.3	利用方法	3
2.4	関数詳細	5
2.4.1	Authenticate	5
2.4.2	setServiceInterface	6
2.4.3	getCurrentUserName	7
2.4.4	getCurrentUserID	8
2.4.5	getCurrentUserOperationAuthority	9

# 1. 概要

## 1.1 概要

認証ライブラリは、V-T500/V-N500 の認証に関するライブラリです。  
V-T500/V-N500 に搭載されている認証サービスのインタフェースとして動作します。

## 1.2 動作環境

### 1.2.1 対象機種

V-T500/V-N500

### 1.2.2 対象OS

Android 4.0 (API level 15)

### 1.2.3 プログラミング言語

Java (JDK コンパイラ準拠レベル 16)

### 1.2.4 必要なシステム

開発環境に必要なシステムにつきましては、ファーストステップガイドをご参照下さい。

---

## 1.2.5 必要なソフトウェア

以下のソフトウェアが必須となります。

- Java SE Development Kit 6u31
- Eclipse 3.7 (Indigo)
- Android SDK r18
- ADT Plugin for Eclipse 15.0.0 以上

## 2. ライブラリ

### 2.1 Package名称

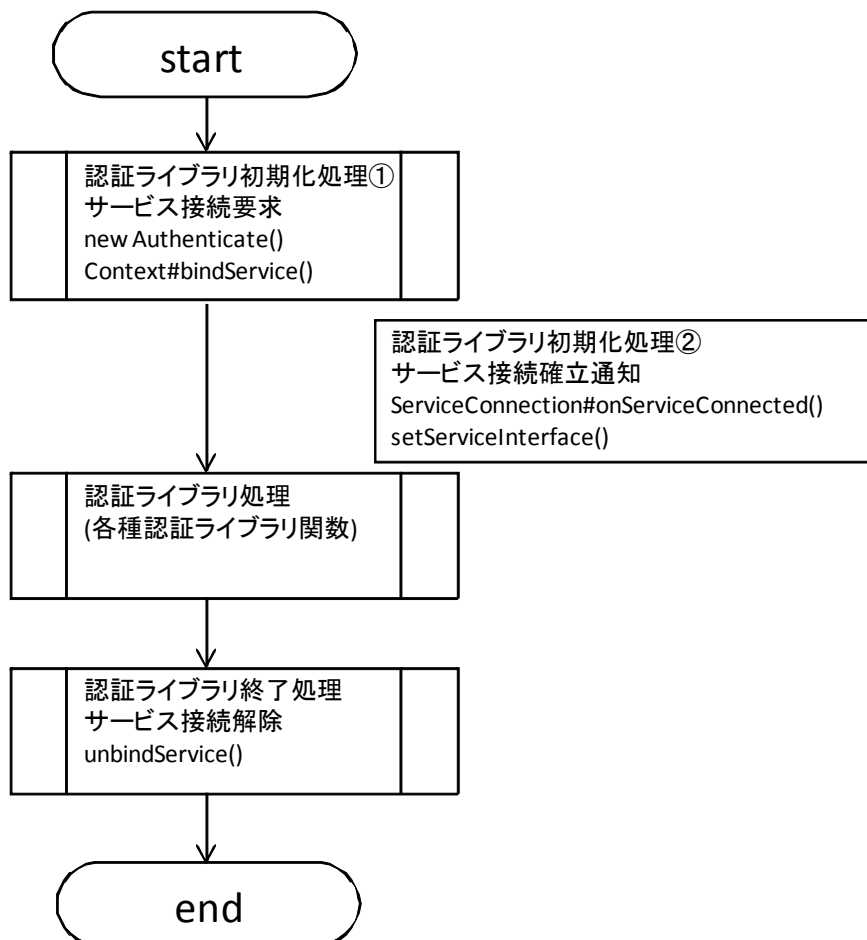
jp.casio.vx.framework.authenticate

### 2.2 関数一覧

関数名	機能
Authenticate	コンストラクタ
setServiceInterface	認証ライブラリにサービスインタフェースを設定します。
getCurrentUserName	現在ログインしているユーザー名を取得します。
getCurrentUserID	現在ログインしているユーザーIDを取得します。
getCurrentUserOperationAuthority	現在ログインしているユーザー業務権限を取得します。

### 2.3 利用方法

認証ライブラリは、ユーザーパスワードおよび管理者パスワードを管理する `android.app.Service` クラスを継承して実装している認証サービスに接続して処理を行います。そのため、認証ライブラリを使用するアプリケーションは、`android.app.Service` 接続処理(初期化処理①、初期化処理②)を実施後に各種認証ライブラリ関数を利用してください。また、最後に終了処理としてサービスから切断処理を実施する必要があります。



以下に、認証ライブラリ使用のサンプルを記載します。

```
import jp.casio.vx.framework.activemenu.authenticate.service.IVXAuthenticateService;
import jp.casio.vx.framework.authenticate.Authenticate;
:
:
private Authenticate mVXAuthenticate;

public void onCreate(Bundle savedInstanceState){
:
:
// 認証ライブラリ初期化処理①(オブジェクト生成、認証サービス接続処理)
mAuthenticate = new Authenticate();

Intent intent = new Intent(IVXAuthenticateService.class.getName());
if(!bindService(intent, mServiceConn,Context.BIND_AUTO_CREATE)){
:
return;
}
}

private ServiceConnection mServiceConn = new ServiceConnection() {
@Override
public void onServiceConnected(ComponentName name, IBinder service) {
IVXAuthenticateService ServiceIf;
ServiceIf = IVXAuthenticateService.Stub.asInterface(service);

// 認証ライブラリ初期化処理②
// 認証サービスと接続ができた段階で、サービス Interface を認証ライブラリに登録
// 登録後、認証ライブラリの各種関数が利用可能になります。
mAuthenticate.setServiceInterface(ServiceIf);

// 認証サービスの利用
// setServiceInterface() 呼び出し後であれば他のメソッドでも利用可能
mAuthenticate.****();
}
@Override
public void onServiceDisconnected(ComponentName arg0) {
:
}
}

public void onDestroy(){
// 認証ライブラリ終了処理
// 認証ライブラリ使用終了後に接続解除処理を行います。
unbindService(mServiceConn);
}
}
```

## 2.4 関数詳細

### 2.4.1 Authenticate

認証ライブラリオブジェクトを構築します。

```
public void Authenticate()
```

#### 解説

本関数は認証ライブラリのコンストラクタです。認証ライブラリオブジェクトを構築します。

#### パラメータ

なし

#### 戻り値

なし

#### 例外

なし

#### 補足

認証ライブラリの初期化処理に使用します。



## 2.4.2 setServiceInterface

認証ライブラリオブジェクト構築後にサービスインタフェースを設定します。

```
public void setServiceInterface(  
    IVXAuthenticateService serviceIf  
)
```

### 解説

本関数は、認証ライブラリ構築後、認証サービスのサービスインタフェースを登録します。  
認証ライブラリの関数は、本関数でサービスインタフェース登録後に使用できます。

### パラメータ

*serviceIf*

認証サービスのサービスインタフェースを指定します。

### 戻り値

なし

### 例外

なし

### 補足

認証ライブラリの初期化処理に使用します。

## 2.4.3 getCurrentUserName

本関数は、現在ログインしているユーザー名を取得します。

```
public String getCurrentUserName ()
```

### 解説

現在ログインしているユーザーのユーザー名を取得します。

### パラメータ

なし

### 戻り値

以下の値を返します。

String	成功(ユーザー名)
空文字列	失敗

### 例外

関数未サポート時に以下の例外を投げます。

**FunctionNotFoundException**

### 補足

ユーザー認証(業務メニュー)でログインしたユーザー情報が表示されます。

## 2.4.4 getCurrentUserID

本関数は、現在ログインしているユーザーIDを取得します。

```
public String getCurrentUserID ()
```

### 解説

現在ログインしているユーザーのユーザーIDを取得します。

### パラメータ

なし

### 戻り値

以下の値を返します。

String	成功(ユーザーID)
空文字列	失敗

### 例外

関数未サポート時に以下の例外を投げます。

`FunctionNotFoundException`

### 補足

ユーザー認証(業務メニュー)でログインしたユーザー情報が表示されます。

## 2.4.5 getCurrentUserOperationAuthority

本関数は、現在ログインしているユーザーの業務権限を取得します。

```
public String getCurrentUserOperationAuthority ()
```

### 解説

現在ログインしているユーザーの業務権限を取得します。

### パラメータ

なし

### 戻り値

以下の値を返します。

String	成功(業務権限)
空文字列	失敗

### 例外

関数未サポート時に以下の例外を投げます。

FunctionNotFoundException

### 補足

ユーザー認証(業務メニュー)でログインしたユーザー情報が表示されます。

## カシオ計算機お問い合わせ窓口

### 製品に関する最新情報

- 製品サポートサイト（カシオペア・ハンディターミナル）

<http://casio.jp/support/ht/>

### 製品の取扱い方法のお問い合わせ

- 情報機器コールセンター



**0570-022066**

市内通話料金でご利用いただけます。

携帯電話・PHS 等をご利用の場合、**042-503-7241**

**カシオ計算機株式会社**

〒151-8543 東京都渋谷区本町 1-6-2

TEL 03-5334-4637(代)