

## *PanelService v1 開発者ガイド*

*SDK/J Authentication Package Version:1.0*



## 重要

---

Copyright © 2008 Ricoh Co., Ltd.

## ご注意

1. 本書の内容に関しては、将来予告無しに変更することがあります。
2. 本書の一部または全部を無断で複写、複製、改変、引用、転載、配布することはできません。
3. 本書を使用したことによるお客様の損害、遺失利益、または第三者のいかなる請求につきましても、当社は一切その責任を負いかねますので、あらかじめご了承ください。

### 4. 商標について

Ethernetは、富士ゼロックス株式会社の登録商標です。

PostScript、Acrobatは、アドビシステムズ社の各国での登録商標または商標です。

Windowsは、米国Microsoft Corporationの米国及びその他の国における登録商標です。

UNIXは、X/Openカンパニーリミテッドがライセンスしている米国ならびに他の国における登録商標です。

Red Hatは、Red Hat, Inc.の米国およびその他の国における商標または登録商標です。

Linuxは、Linus Torvalds氏の米国およびその他の国における商標または登録商標です。

Java、JavaCard、JVM (CVM)、CDCは、すべてSUNの米国およびその他の国における商標または登録商標です。

Eclipseは、eclipse.orgの米国およびその他の国における商標または登録商標です。

OSGi(TM)はThe Open Services Gateway Initiativeの米国およびその他の国における商標または登録商標です。

Apacheは、The Apache Software Foundationの米国およびその他の国における商標または登録商標です。

その他の製品名、名称は、各社の商標または登録商標です。

## 目次

<b>1. はじめに</b>	<b>2</b>
1.1. 対象読者	2
1.2. 動作環境	2
1.3. 制限事項	2
<b>2. 概要</b>	<b>3</b>
2.1. Ccm クラス	5
2.2. CcmService クラス	5
2.3. CcmRequestHandler インタフェース	5
2.4. CcmRequest インタフェース	5
<b>3. 画面遷移図</b>	<b>6</b>
<b>4. アプリケーションの開発</b>	<b>7</b>
4.1. CcmService の取得	7
4.2. requestLogin ( )、requestLogout ( ) の実装	7
4.3. CcmRequestHandler インタフェースの実装と CcmService へのセット/削除	8
<b>5. アプリケーションのインストール</b>	<b>10</b>
5.1. DALP ファイルの設定	10
5.2. インストール	10
<b>変更履歴</b>	<b>11</b>

# 1. はじめに

本書は、Panel Service v1.0 (以下、Panel Service) の使用方法について説明します。Panel Service は、Device SDK Type-J (以下、SDK/J) で、本体システムのユーザ認証機能を利用するためのソフトウェアフレームワークです。

## 1.1. 対象読者

本書は、Panel Service を使用する SDK/J アプリケーションの開発者を対象に記述されております。本書を読み進めるにあたり、以下の知識が必要になります。

- SDK/J アプリケーションを開発するための十分な知識
- 本体システムのアクセスロール機能に関する十分な知識

なお、Panel Service を使用したアプリケーションの使用に際しては、本体の認証機能の設定が必要です。詳細は、「インストール手順書」を参照ください。

## 1.2. 動作環境

Panel Service を使用するには、以下の環境が必要となります。

- Panel Service が、有効に設定されている  
(Panel Service の設定方法は、「SDK/J Authentication Package 設定ガイド」を参照ください。)

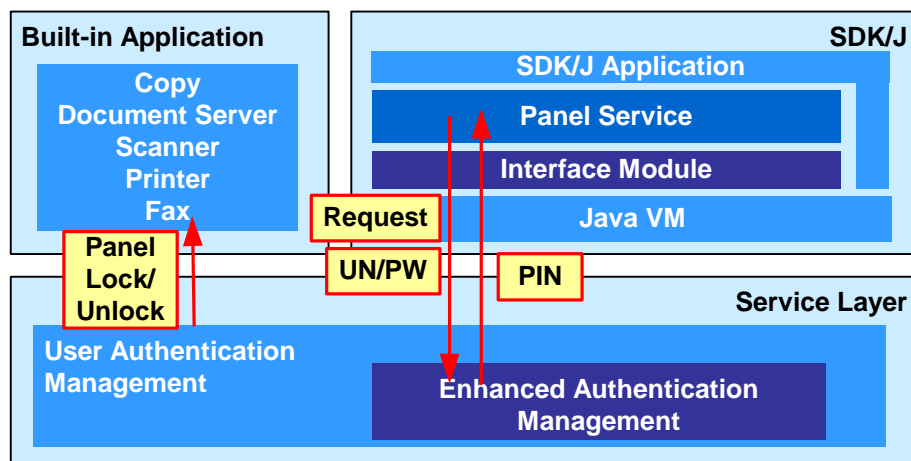
## 1.3. 制限事項

Panel Service には、以下の制限事項があります。

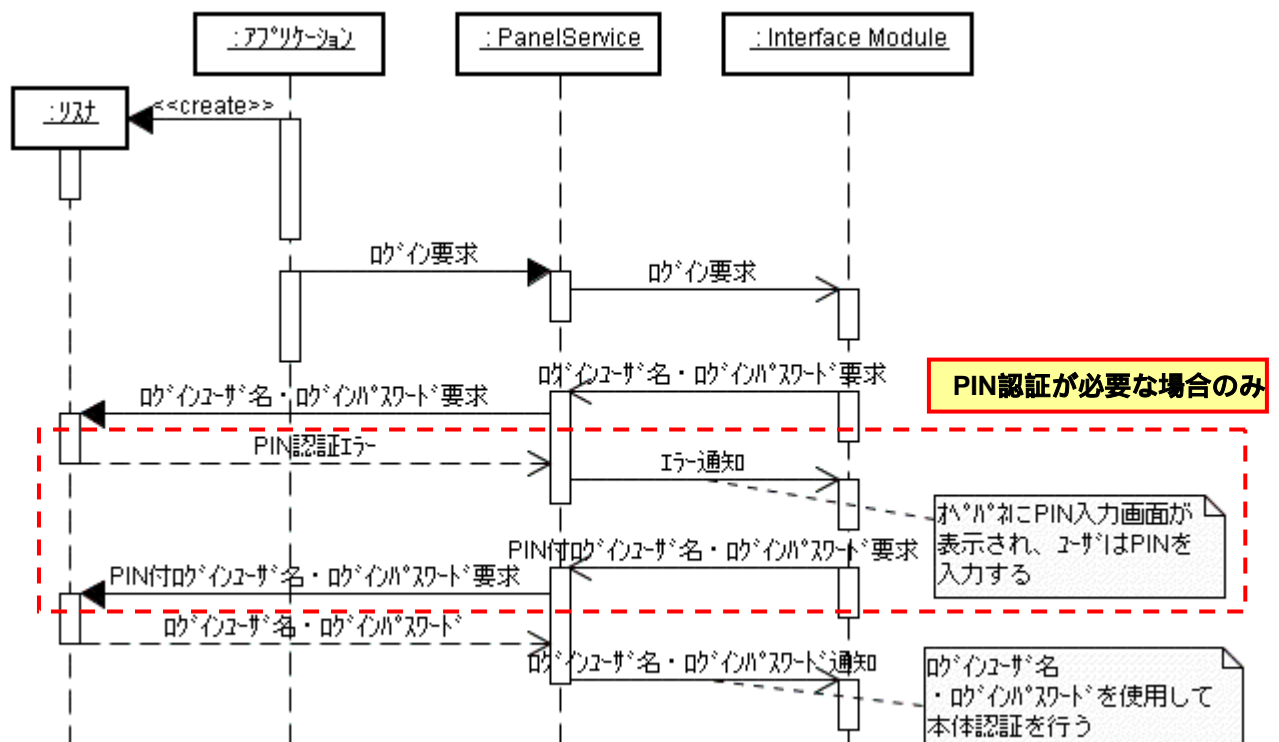
- Panel Service を複数のアプリケーションから同時に利用することは出来ません。Panel Service を利用したアプリケーションを複数起動した場合、最初に起動したアプリケーションのみが Panel Service を利用できません。
- CcmRequestHandler の handle メソッド内の処理は 30 秒以内に終了する必要があります。30 秒以内にメソッドが終了しない場合、本体ユーザ認証モジュールの仕様により認証失敗となります。(厳密には、ユーザ認証モジュールからのログインユーザ名・ログインパスワード要求およびPIN認証要求に対するPanel Service からのレスポンスが 30 秒以内という仕様になります。)
- Panel Service では、各種ユーザ認証画面をカスタマイズすることはできません。
- Panel Service で使用可能なユーザ名・パスワード文字および文字数制限は、機器のアクセスロール仕様に依存します。詳細は、機器のユーザガイドを参照ください。また、PIN は、1~8 文字の数字である必要があります。
- Panel Service によるログイン要求は、認証要求画面が操作パネル上に表示されている場合のみ有効です。
- Panel Service によるログアウト機能を使用するには、本体システムファームウェアが拡張認証によるログアウトをサポートしている必要があります。詳細は、RiDP にお問い合わせください。

## 2. 概要

Panel Service は、SDK/J アプリケーションからログインユーザ名・ログインパスワードを利用した本体ユーザ認証（ベーシック認証、Windows認証、LDAP認証、統合サーバー認証）の解除を行うためのライブラリです。Panel Service の構成要素およびそれに関連するソフトウェアの配置を簡略化した図を、以下に示します。Panel Service は、Authentication Package Bundle の提供する Interface Module を介して本体ユーザ認証モジュールとログインユーザ名・ログインパスワード・PINなどの認証パラメータの受け渡しを行い、本体のユーザ認証を解除します。

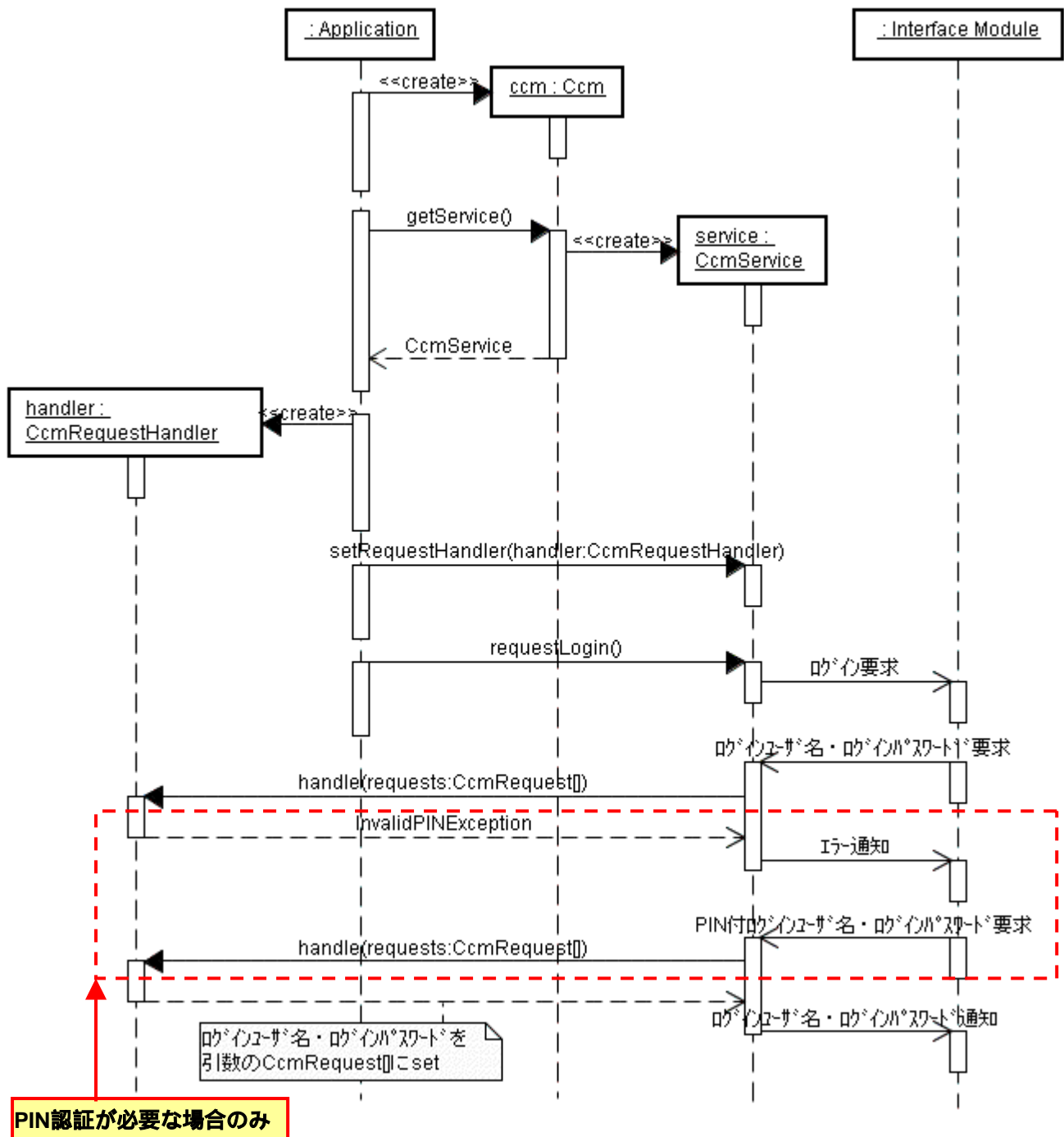


以下の図は、Panel Service のログインの際の動作シーケンスを表しています。アプリケーションは、例えば認証に使用するスマートカードがカードリーダに挿入された際などに、Panel Service にログイン要求を行います。Panel Service が Interface Module にログイン要求を伝えると、ログインユーザ名・ログインパスワード要求が返されます。これらの要求は、アプリケーションがあらかじめ登録したリスナによって処理されます。ログインユーザ名・ログインパスワードの取得に PIN 認証が必要な場合は、Interface Module に PIN 認証エラーを通知することにより、オペパネ上にユーザからの PIN 入力を受け付ける PIN 入力画面を表示できます。



以上のシーケンスを、Panel Service のクラスを使用した実際のシーケンスで表すと、以下になります。アプリケーションはフロントエンドである Ccm オブジェクトを利用して CcmService オブジェクトを取得し、requestLogin() メソッドを利用してログイン要求を行います。Interface Module から受信するログインユーザ名・ログインパスワード要求 (CcmRequest) は、あらかじめ CcmService オブジェクトに登録した CcmRequestHandler 実装オブジェクトの handle(CcmRequest[]) メソッドを利用して処理します。

PIN認証エラーが発生した際は、handle メソッド から InvalidPINException をスローすることで、オペパネ上にPIN入力画面が表示されます。



以下に、Panel Service の実装を行う上で主要となるクラスについて、簡単に説明します。詳細は、Javadoc を参照ください。

## 2.1. Ccm クラス

アプリケーションに対するフロントエンドとなります。getService() メソッドにより、CcmService のインスタンスを取得することが出来ます。

## 2.2. CcmService クラス

ユーザ認証モジュールへの I/F となるクラスです。requestLogin()、requestLogout() メソッドにより、ログイン要求、ログイン要求キャンセル要求/ログアウト要求を Interface Module を介して認証モジュールに伝えます。

また、setRequestHandler() メソッドにより登録された CcmRequestHandler 実装オブジェクトを利用して、認証モジュールからの要求を処理します。CcmRequestHandler 実装オブジェクトは、1つまでしか登録できないことに注意してください。

## 2.3. CcmRequestHandler インタフェース

ユーザ認証モジュールからの要求に対する処理を行う handle() メソッドを定義したインタフェースです。このインタフェースは、Panel Service アプリケーション開発者が実装する必要があります。

## 2.4. CcmRequest インタフェース

ユーザ認証モジュールからの要求であることを表すインタフェースです。

Panel Service では、このインタフェースの実装として、ログインユーザ名を表す Name クラス、ログインパスワードを表す Password クラス、PIN認証要求を表す VerifyPIN クラスを使用しています。





## 4. アプリケーションの開発

この章では、簡単なサンプルコードを用いて、Panel Service を利用したアプリケーションの実装方法を説明します。

Panel Service アプリケーションの開発には、**panel-service.jar** 内のクラスを使用します。

なお、SDK/J アプリケーションの作成に関する内容や、厳密な例外処理については省略します。

### 4.1. CcmService の取得

Panel Service のフロントエンドクラスである Ccm クラスのインスタンスを生成して、getService() メソッドにより CcmService のインスタンスを取得します。

```

・
・
/** CcmService のインスタンスを取得 */
CcmService ccmService = new Ccm().getService();
・
・

```

### 4.2. requestLogin ( )、requestLogout ( ) の実装

ログイン要求およびログイン要求キャンセル/ログアウト要求をユーザ認証モジュールに伝えるメソッドである requestLogin()、requestLogout() メソッドが、適切なタイミングで呼び出されるように実装します。

下記サンプルコードでは、カードがカードリーダーに挿入された際に requestLogin() メソッドを、カードがカードリーダーから除去された際に requestLogout() メソッドを呼び出しています。

```

・
・
CardManager cardManager = new CardManager();

cardManager.addCardListener(
    new CardEventListener() {
        /** カード挿入時に、ログイン要求を本体認証モジュールに伝える */
        public void inserted(CardEvent e) {
            ccmService.requestLogin();
        }

        /** カード除去時に、ログイン要求キャンセル/ログアウト要求を本体認証モジュールに伝える */
        public void removed(CardEvent e) {
            ccmService.requestLogout();
        }
    }
);
・
・

```

### 4.3. CcmRequestHandler インタフェースの実装と CcmService へのセット/削除

ログイン処理の中で、ユーザ認証モジュールからの Panel Service への要求 (CcmRequest) を処理するためのインタフェースである CcmRequestHandler を実装します。

handle() メソッドで、各種 CcmRequest (Name または Password または VerifyPIN) に対する処理を実装していきます。まずCcmRequestを振り分け、その後各リクエストを処理する、という手順で実装します。

処理中に PIN 認証に関するエラーが発生した場合には、handle() メソッドから InvalidPINException をスローして、ユーザ認証モジュールに PIN 認証エラーが発生したことを通知する必要があります。

```

        .
        .
CcmRequestHandler myHandler = new CcmRequestHandler() {
    /** handle メソッドを実装する */
    public void handle(CcmRequest[] requests)
        throws UnsupportedOperationException, InvalidPINException, IOException {
        Name name = null;
        Password password = null;
        VerifyPIN pin = null;

        /** リクエスト振り分け */
        for (int i = 0; i < requests.length; i++) {
            if (requests[i] instanceof Name) {
                name = (Name) requests[i];
            } else if (requests[i] instanceof Password) {
                password = (Password) requests[i];
            } else if (requests[i] instanceof verifyPIN) {
                pin = (VerifyPIN) requests[i];
            } else {
                /** 不正なリクエストの場合 */
                throw new UnsupportedOperationException();
            }
        }

        /** 各リクエストの処理 */
        try {
            if (pin != null) {
                byte[] pinBuffer = pin.getPIN();
                /** PIN認証 */
                cardService.verifyPIN(pinBuffer);
            }
            if (name != null) {
                /** ログインユーザ名の取得 */
                byte [] nameBytes = cardService.getUser();
                name.setName(nameBytes);
            }
            if (password != null) {
                /** ログインパスワードの取得 */
                byte [] passBytes = cardService.getPassword();
                password.setPassword(passBytes);
            }
        } catch (CHVException e) {
            /** PIN認証エラーがあった場合 */
            throw new InvalidPINException();
        }
    }
};
        .
        .

```

次に、実装した `CcmRequestHandler` が Panel Service からコールバックされるために、`setRequestHandler()` メソッドを使用して `CcmService` インスタンスにセットします。これにより、ユーザ認証モジュールからの要求はセットされた `CcmRequestHandler` で処理されるようになります。`CcmService` にセットできる `CcmRequestHandler` は 1 つまでです。既に `CcmRequestHandler` がセットされている場合、`AlreadySetException` がスローされます。

```
.  
.br/>/** CcmRequestHandler をセットする */  
ccmService.setRequestHandler(myHandler);  
.br/.
```

なお、アプリケーションの終了時には、`removeRequestHandler()` メソッドを使用し、セットした `CcmRequestHandler` を削除するようにしてください。

```
.  
.br/>/** セットした CcmRequestHandler を削除する */  
ccmService.removeRequestHandler(myHandler);  
.br/.
```

## 5. アプリケーションのインストール

Panel Service を利用した SDK/J アプリケーションのインストール方法は、以下になります。

### 5.1. DALP ファイルの設定

DALP ファイルの書式に関しては、SDK/J 開発者ガイドを参照してください。

Panel Service の jar ファイル (panel-service.jar) は、SDK/J の SD カードに同梱されているので、panel-service.jar を DALP ファイル内で指定する必要はありません。

### 5.2. インストール

通常の SDK/J アプリケーションと同様にインストールしてください。

詳細は、SDK/J のユーザズガイドを参照ください。

Panel Service は MFP/LP ファームウェア内のユーザ認証モジュールと連携して動作するため、エミュレータによる動作確認を行うことはできません。

なお、Panel Service の利用にあたっては MFP/LP の認証機能の設定を行う必要があります。詳細は、「Panel Service v1 ユーザズガイド」を参照ください。

## 変更履歴

Ver. 1.0	SDK/J v2 オプションパッケージ版 v1.0 をベースに SDK/J v4 以降版を作成 動作環境 を更新 Implementation-Version: 1.1-1.1
----------	--