

公的個人認証サービス

利用者クライアントソフト API 仕様書 【Android インテント編】

第 1.7 版

地方公共団体情報システム機構

変更履歴

版数	変更日付	変更内容
1.0 版	平成 28 年 8 月 31 日	新規作成
1.1 版	平成 29 年 7 月 31 日	<p>Java9 対応に伴い、以下を修正。</p> <ul style="list-style-type: none"> 第 2 章 ドキュメント体系図を改訂。 第 3 章 第 1 節 表 3 - 1 の【PC 接続の場合】に Android 6.0.1、7.0 を追加。 第 3 章 第 1 節 表 3 - 1 の【Android 単体で利用する場合】に Android 6.0.1、7.0 を追加。 第 5 章 第 1 節 表 5 - 3 のリザルトコード : 21 の意味を修正。
1.2 版	平成 30 年 8 月 31 日	<p>Android SDK(26)対応に伴い、以下を修正。</p> <ul style="list-style-type: none"> 第 3 章 第 1 節 表 3 - 1 の【PC 接続の場合】に Android 8.0 を追加。 第 3 章 第 1 節 表 3 - 1 の【Android 単体で利用する場合】に Android 8.0 を追加。
1.3 版	令和元年 5 月 1 日	<p>新元号対応に伴い、以下を修正。</p> <ul style="list-style-type: none"> 第 3 章 第 1 節 JPKI 利用者ソフトのバージョンを 1.1 に変更。 第 3 章 第 2 節 JPKI 利用者ソフトのバージョンを 1.1 に変更。 第 6 章 第 1 1 節 2 生年月日の設定ルールに新元号を追加。 その他、記載の見直しを実施。 <p>旧氏対応に伴い、以下を修正。</p> <ul style="list-style-type: none"> 第 5 章 第 2 節に旧氏に関する記述を追加。 第 6 章 第 2 節 表 6 - 2 に旧氏に関する記述を追加。

版数	変更日付	変更内容
1.4 版	令和 2 年 3 月 31 日	<p>MacOS 版における住基カードサポート廃止に伴い、以下を修正。</p> <ul style="list-style-type: none"> ・ 第 2 章 ドキュメント体系 「API 仕様書 Mac OS X C 言語インターフェース編」を削除。 <p>ブラウザ対応版、iOS 版のリリースに伴い、以下を修正。</p> <ul style="list-style-type: none"> ・ 第 2 章 ドキュメント体系 「利用者クライアントソフト 機能概要説明書 ブラウザ対応編」を追加。 ・ 第 2 章 ドキュメント体系 「API 仕様書 カード AP ライブラリ ブラウザインターフェース編」を追加。 ・ 第 2 章 ドキュメント体系 「利用者クライアントソフト 機能概要説明書(iOS 版)」を追加。 ・ 第 2 章 ドキュメント体系 「API 仕様書 iOS Framework 編」を追加。 ・ 第 3 章 第 1 節 動作環境に Android9.0、10.0 を追加。
1.5 版	令和 3 年 3 月 31 日	<p>ブラウザ対応版(Android)のリリースに伴い、以下を修正</p> <ul style="list-style-type: none"> ・ 第 2 章 ドキュメント体系 「利用者クライアントソフト 機能概要説明書(Android 版) ブラウザ対応編」を追加。 ・ 第 2 章 ドキュメント体系 「API 仕様書 Android インテント ブラウザインターフェース編」を追加。
1.5.1 版	令和 5 年 1 月 31 日	<p>Android SDK31 対応に伴い、以下を修正</p> <ul style="list-style-type: none"> ・ 第 3 章 第 1 節 Android 端末 JPKI 利用者ソフトのバージョンを 2.0.2 に変更 ・ 第 3 章 第 1 節 Android 端末 OS から Android4.3、5.1、6.0.1 を削除し、Android11.0、12.0 を追加 ・ 第 3 章 第 2 節 前提条件 JPKI 利用者ソフトのバージョンを 2.0.2 に変更
1.6 版	令和 5 年 3 月 31 日	<p>利用者同意状況確認機能追加に伴い、以下を修正</p> <ul style="list-style-type: none"> ・ 第 1 章 第 1 節 用語の定義の説明文を変更 ・ 第 3 章 第 1 節 JPKI 利用者ソフトのバージョンを 2.1 に変更。 ・ 第 3 章 第 1 節 表 3-1 Android 7.0 を削除。 ・ 第 6 章 第 9 節 パスワード入力画面の画面文言を修正 ・ 第 6 章 第 10 節 IC カードセット案内画面の画面文言を修正

版数	変更日付	変更内容
1.6.2 版	令和 5 年 3 月 31 日	以下を修正 <ul style="list-style-type: none"> ・第 3 章 動作環境の記載を修正
1.7 版	令和 6 年 7 月 31 日	Android SDK34 対応に伴い、以下を修正 <ul style="list-style-type: none"> ・第 2 章 ドキュメント体系 図 2 - 3 ドキュメント体系図(iOS 版)に API 仕様書 iOS XCFramework 編を追加 ・第 3 章 第 1 節 Android 端末 JPKI 利用者ソフトのバージョンを 2.2 に変更 ・第 3 章 第 1 節 Android 端末 OS 条件に Android13.0、14.0 を追加 ・第 3 章 第 2 節 前提条件 JPKI 利用者ソフトのバージョンを 2.2 に変更 ・巻末 注意事項欄の商標表記を Google LLC に変更

- 目次 -

第1章 はじめに	1-1
第1節 用語の定義	2
第2章 ドキュメント体系	3-3
第3章 動作環境	5-5
第1節 Android 端末	5-5
第2節 前提条件	6-6
第4章 機能仕様	7-7
第1節 ソフトウェア構成図	7-7
第2節 実現可能な機能の一覧	8-8
第5章 インテント仕様	9-9
第1節 インテントインターフェースの概念図	9-9
1 送信データ	10-10
2 受信データ	11-11
第2節 拡張データに格納する項目一覧	13-13
1 初期化处理(カード AP ライブラリ、個人認証サービス AP 共通)	13-13
2 終了処理(カード AP ライブラリ、個人認証サービス AP 共通)	14-14
3 カード AP ライブラリ(署名用の場合)	15-15
4 カード AP ライブラリ(利用者証明用の場合)	21-21
5 個人認証サービス AP(署名用、利用者証明用共通)	25-25
第3節 コーリングシーケンス	32-32
1 初期処理	32-32
2 終了処理	34-34
3 カード AP ライブラリ(署名用の場合)	36-36
4 カード AP ライブラリ(利用者証明用の場合)	46-46
5 個人認証サービス AP(署名用、利用者証明用共通)	52-52
第6章 画面仕様	60-60
第1節 画面一覧	60-60
第2節 署名用電子証明書基本画面	62-62
第3節 利用者証明用電子証明書基本画面	65-65
第4節 認証局の自己署名証明書基本画面	68-68
第5節 その他の電子証明書基本画面	71-71
第6節 詳細画面	74-74
第7節 官職証明書・職責証明書検証画面	80-80
第8節 利用者証明書有効性確認画面	83-83
第9節 パスワード入力画面	87-87
第10節 IC カードセット案内画面	89-89
第11節 設定ルール一覧	90-90

第 1 章 はじめに

公的個人認証サービス利用者クライアントソフト(以下、JPKI 利用者ソフト)は、以下の機能を実現するためのインテントインターフェースを提供する。

- 証明書取得機能
- 電子署名生成機能
- 電子署名検証機能
- 証明書表示機能
- 基本 4 情報取得機能
- 官職証明書検証機能
- 自分の電子証明書の有効性確認機能
- IC カード種別取得機能

以降、本書では Android 版 JPKI 利用者ソフトのインテントインターフェースの仕様について説明する。

第 1 節 用語の定義

表 1 - 1 用語の定義

#	用語・略号	説明
1	IC カード	以下のカードを指す総称 ・ 個人番号カード
2	電子証明書	公開鍵及び発行対象を識別する情報を含むデータに、認証局が発行対象の正当性を保証する電子署名を付与し、発行されるデータを示す。 データは、ISO/IEC 8825-1 の識別符号化規則により符号化された形式で利用される。
3	証明書	電子証明書と同義
4	利用者証明書	公的個人認証サービスで発行した利用者の証明書 具体的には以下の電子証明書を示す。 ・ 個人番号カードに格納された署名用電子証明書 ・ 個人番号カードに格納された利用者証明用電子証明書 ・ 移動端末設備に格納された署名用電子証明書 ・ 移動端末設備に格納された利用者証明用電子証明書
5	認証局の自己署名証明書	自認証局の公開鍵に対して、自認証局の秘密鍵で署名した証明書。 本書では以下の電子証明書を示す。 ・ 個人番号カードに格納された署名用認証局の自己署名証明書 ・ 個人番号カードに格納された利用者証明用認証局の自己署名証明書 ・ 移動端末設備に格納された署名用認証局の自己署名証明書 ・ 移動端末設備に格納された利用者証明用認証局の自己署名証明書
6	NFC	Near Field Communication (近距離無線通信)の略。
7	インテント	Android の機能の一つ。アプリケーション間やアプリケーション内の機能間において、情報を連携するための機能。

第2章 ドキュメント体系

JPKI 利用者ソフトのドキュメント体系図を以下に示す。本書は以下の体系図の網掛け部分に該当する。

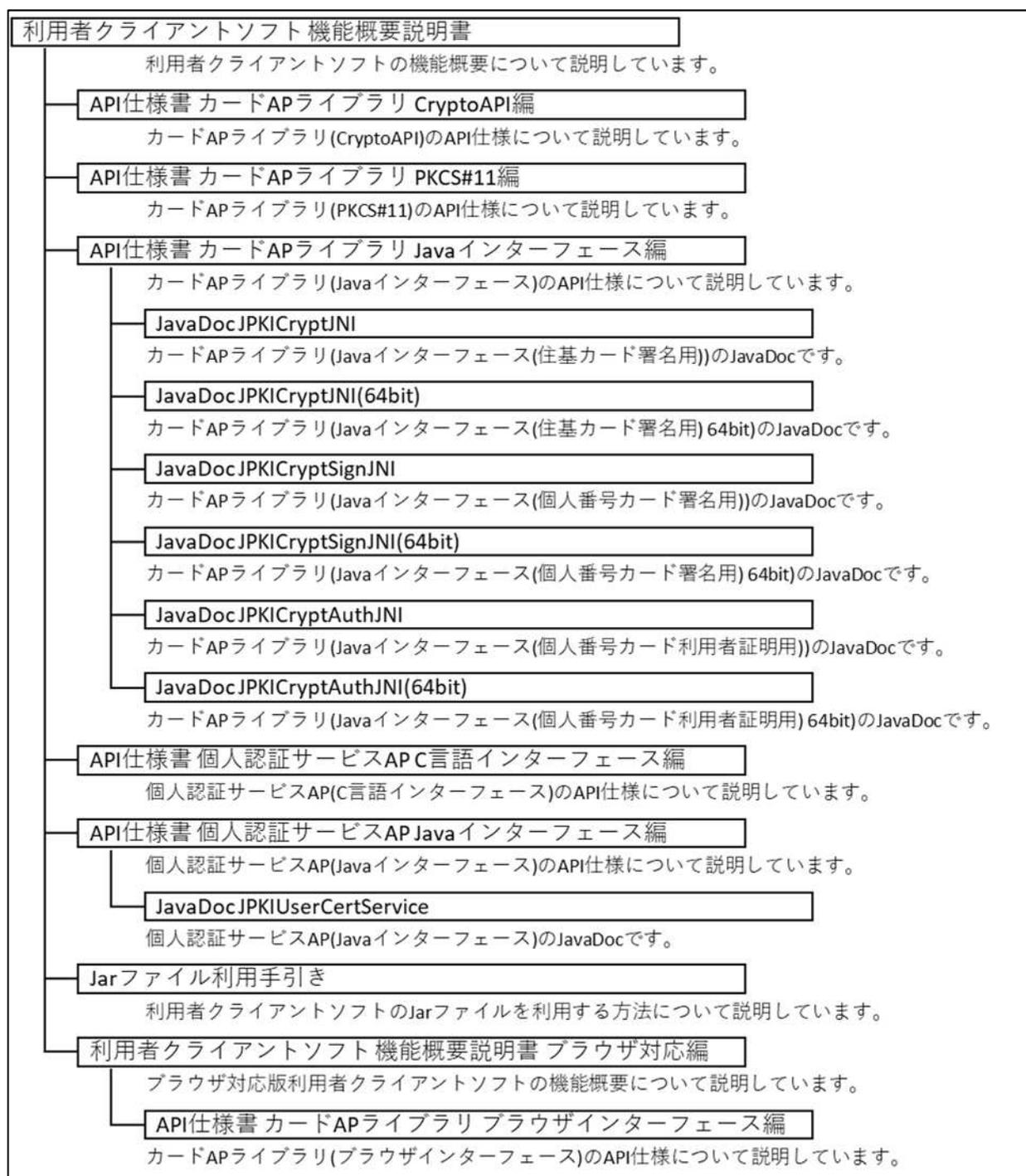


図 2-1 ドキュメント体系図(PC版)

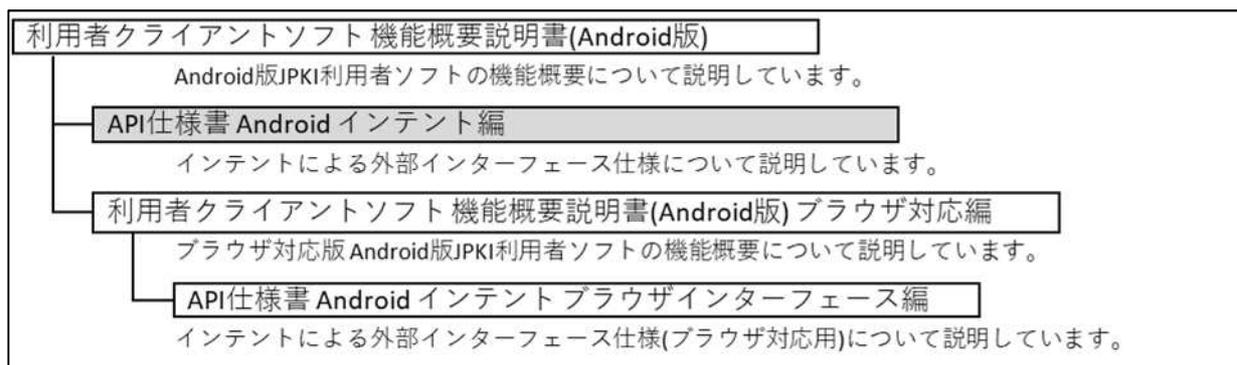


図 2-1-2 ドキュメント体系図(Android版)

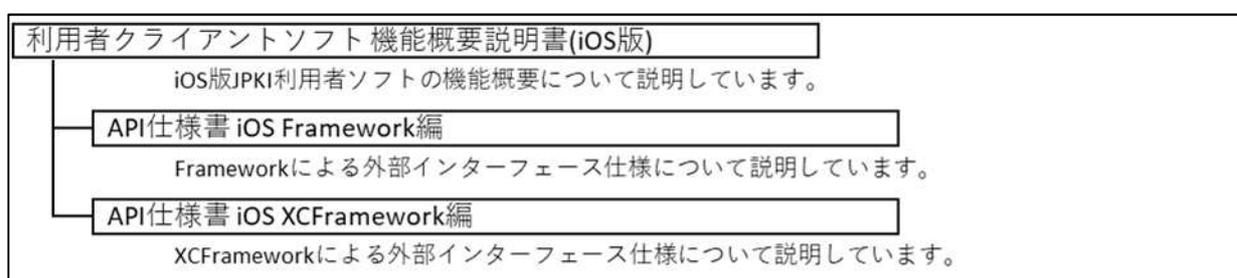


図 2-2-3 ドキュメント体系図(iOS版)

第 3 章 動作環境

第 1 節 Android 端末

Android 版 JPKI 利用者ソフト Ver2.2 の動作環境は以下の通りとする。

表 3-1 動作環境

項目	条件
OS	下記の Android のいずれかを搭載していること。 ・ 8.0、 9.0、 10.0、 11.0、 12.0、 13.0、 14.0
NFC	以下の条件を満たす Android 端末とする。（「個人番号カード対応適合性検証済み Android 端末一覧」 ¹ を参照のこと。） ・ ISO/IEC 14443 Type B に対応している NFC を搭載していること。
IC カード	個人番号カードであること。（住基カードは対象外）

1 最新の「個人番号カード対応適合性検証済み Android 端末一覧」の情報は、JPKI ポータルサイトに掲載するものとする。

また、最新の「個人番号カード対応適合性検証済み Android 端末」以外を使用した場合の不具合等に関するお問い合わせは、サポート対象外とする。

表示画面については以下の通りとする。

表 3-2 表示画面

項目	仕様
解像度	フル HD または HD
画面比	16:9
画面の向き	縦画面固定とする。

Android 版 JPKI 利用者ソフトにおける Android 端末のボタン操作について以下の通りとする。

表 3-3 ボタン操作

ボタン名	動作
ホームボタン	使用しない。
履歴ボタン	使用しない。
戻るボタン	1 つ前の画面に戻る。

第2節 前提条件

Android 版 JPKI 利用者ソフト Ver2.2 の前提条件は以下の通りとする。

表 3-4 前提条件

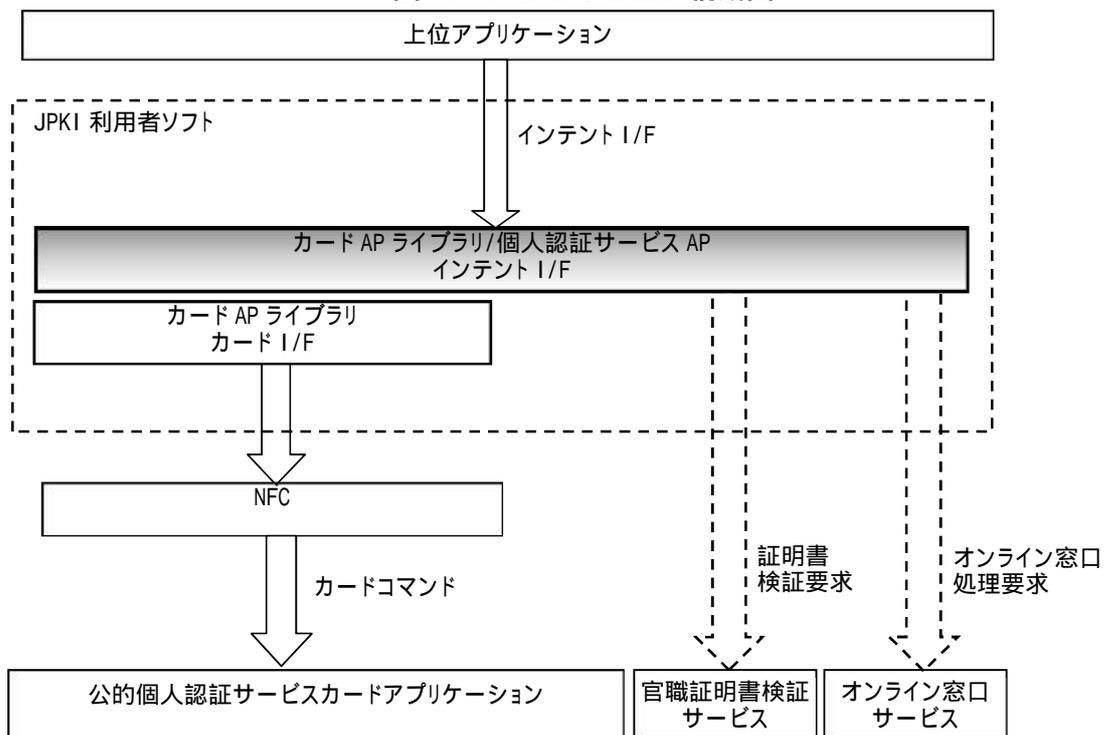
前提条件
Android 版 JPKI 利用者ソフトで設定、登録した情報は機種変更等による Android 端末間の移行には対応しない。
Android のマルチユーザ機能には対応しない。

第4章 機能仕様

第1節 ソフトウェア構成図

本仕様書では、Android版 JPKI 利用者ソフトのうち、下図の太枠に示すインテントインターフェースの仕様をまとめる。

図 4-1 ソフトウェア構成図



第2節 実現可能な機能の一覧

インテントで実現可能な機能の一覧を以下に示す。

表 4 - 1 実現可能な機能の一覧

NO	カテゴリ	機能	概要
1	カード AP ライブラリ	証明書取得	IC カードに格納された電子証明書(利用者証明書、認証局の自己署名証明書)を取得する。
2		署名生成	署名対象データからハッシュ値を計算し、IC カードに格納された利用者秘密鍵を使用して電子署名を生成する。
3		署名検証	検証対象データからハッシュ値を計算し、ハッシュ値、電子署名、公開鍵を使用して電子署名を検証する。
4	個人認証サービス AP	証明書表示	電子証明書を表示する。
5		基本 4 情報取得	署名用電子証明書から基本 4 情報(氏名、住所、性別、生年月日)を取得する。
6		官職証明書検証	官職証明書や職責証明書の証明書検証を行うため、公的個人認証サービスの官職証明書検証サービスに対して証明書検証要求を発行する。
7		自分の電子証明書の有効性確認	IC カード内の自分の電子証明書(利用者証明書)の有効性を確認するために、公的個人認証サービスのオンライン窓口サービスに対して有効性確認要求を発行する。
8		IC カード種別取得	端末にセットされている IC カードの種別を取得する。

第 5 章 インテント仕様

第 1 節 インテントインターフェースの概念図

インテントを使用したデータの流れを以下に示す。

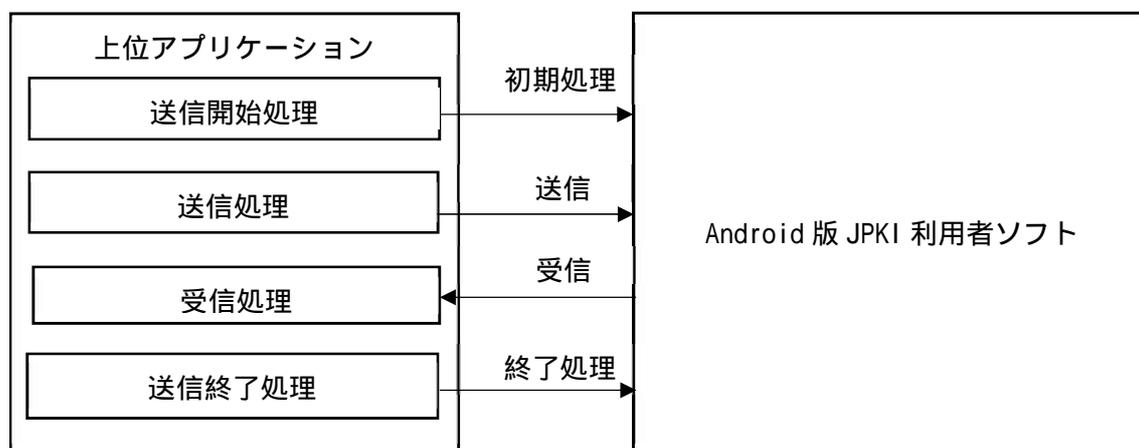


図 5-1 インテントインターフェースの概念図

1 送信データ

図 5-1 図 5-1 において送信するデータについての仕様を以下にまとめる。

上位アプリケーションにおいて `startActivityForResult(Intent intent, int requestCode)` 関数を実行して Android 版 JPKI 利用者ソフトに送信するデータを以下に示す。

表 5-1 送信データ

引数名	項目	説明
intent	intent の動作の指定	<code>Intent.ACTION_SEND</code> を設定すること。
	JPKI 利用者ソフトの パッケージ名とクラス名の設定	以下のように設定すること。 <code>intent.setClassName(</code> <JPKI 利用者ソフトのパッケージ名()>, “ <code>jp.go.jpki.mobile.intent.JPKIIntentActivity</code> ”) ; JPKI 利用者ソフトのパッケージ名には以下の値を使用すること。 一般利用者向け : “ <code>jp.go.jpki.mobile.utility</code> ” 署名検証者向け : “ <code>jp.go.jpki.mobile.utility.dev</code> ”
	拡張データ	<code>intent.putExtra</code> で設定するデータ。 設定項目については第 5 章 第 2 節 を参照すること。
requestCode	リクエストコード	遷移元に戻った際に <code>onActivityResult</code> の引数となる。 数値は上位アプリケーション側で任意の値を設定すること。

2 受信データ

図 5-1 図 5-4 において受信するデータについての仕様を以下にまとめる。

上位アプリケーションにおいて `onActivityResult(int requestCode, int resultCode, Intent data)` で Android 版 JPKI 利用者ソフトから受け取るデータを以下に示す。

表 5-2 受信データ

引数名	項目	説明
<code>requestCode</code>	リクエストコード	送信時に指定したリクエストコードが設定される。
<code>resultCode</code>	リザルトコード	コマンド実行結果が格納される。 値については表 5-3 表 5-3 を参照すること。
<code>data</code>	拡張データ	<code>intent.putExtra</code> で設定するデータ。 設定項目については第 5 章 第 2 節 を参照すること。

表 5-3 リザルトコード一覧

値	意味
0	正常に処理された。
1	実行途中でユーザーがキャンセル操作を行った。
2	指定されたコマンドタイプが存在しない。
3	指定されたコマンドに必要な送信データが不足している。
4	Android 版 JPKI 利用者ソフトで予期しないエラーが発生した。
5	引数に不正なデータを渡した。
6	不正なハッシュハンドルを指定した。
7	ハッシュは既に完了しているためデータを追加できない。
8	不正な鍵ハンドルを指定している。
9	不正なプロバイダハンドルを指定している。
10	不正な鍵情報を指定している。
11	OCSP Request および拡張情報の生成に失敗した。
12	証明書検証サーバの URL が正しくない。
13	証明書(利用者証明書または認証局の自己署名証明書)の取得に失敗した。
14	HTTP または HTTPS 通信で失敗した。
15	証明書検証結果の解析に失敗した。
16	CVS 証明書の有効性検証に失敗した。
17	OCSP Response (ASN.1 エンコード形式) のデコードに失敗した。
18	Request 送信時に付与した nonce と Response に含まれる nonce が一致しない。
19	OCSP レスポンダから受信した OCSP Response (ASN.1 エンコード形式) の署名検証

値	意味
	に失敗した。
20	電子証明書(DER 形式)の解析に失敗した。
21	NFC が無効もしくは、IC カード情報の取得に失敗した (IC カード未挿入等)。
22	IC カードが個人番号カードではないため処理に失敗した。
23	パスワード入力画面でエラーが起こった。
24	IC カードセット案内画面でエラーが起こった。
25	電子証明書表示画面でエラーが起こった。

第 2 節 拡張データに格納する項目一覧

拡張データに格納する項目を以下に示す。

1 初期化処理(カード AP ライブラリ、個人認証サービス AP 共通)

項目名	初期化処理			
概要	最初に必ず実行しなければならない処理			
	キー	型	値	内容
送信データ	command_type	int	0x01004001	コマンドタイプ
受信データ	command_type	int	0x01004001	コマンドタイプ
	result	boolean		初期化処理が成功の場合は true、失敗した場合は false
	err_code	int		エラーコード(4桁) 例外が発生した箇所のエラーコードの値が格納される。 正常系の場合は「0000」が格納される。
	detail_code	int		エラー詳細コード(8桁) 例外が発生した箇所の詳細コードの値が格納される。 正常系の場合は「00000000」が格納される。
備考	特に無し。			

2 終了処理(カード AP ライブラリ、個人認証サービス AP 共通)

項目名	終了処理			
概要	終了時に必ず実行しなければならない処理			
	キー	型	値	内容
送信データ	command_type	int	0x01004002	コマンドタイプ
受信データ	command_type	int	0x01004002	コマンドタイプ
	result	boolean		終了処理が成功の場合は true、失敗した場合は false
	err_code	int		エラーコード(4 桁) 例外が発生した箇所のエラーコードの値が格納される。 正常系の場合は「0000」が格納される。
	detail_code	int		エラー詳細コード(8 桁) 例外が発生した箇所の詳細コードの値が格納される。 正常系の場合は「00000000」が格納される。
備考	特に無し。			

3 カード AP ライブラリ(署名用の場合)

(1) 署名用認証局の自己署名証明書取得処理

項目名	署名用認証局の自己署名証明書取得処理			
概要	署名用認証局の自己署名証明書(DER形式)を取得			
	キー	型	値	内容
送信データ	command_type	int	0x01002001	コマンドタイプ
受信データ	command_type	int	0x01002001	コマンドタイプ
	ca_cert	byte[]		署名用認証局の自己署名証明書(DER形式)
	err_code	int		エラーコード(4桁) 例外が発生した箇所のエラーコードの値が格納される。 正常系の場合は「0000」が格納される。
	detail_code	int		エラー詳細コード(8桁) 例外が発生した箇所の詳細コードの値が格納される。 正常系の場合は「00000000」が格納される。
備考	<p>処理の流れは以下となる。</p> <ul style="list-style-type: none"> ICカードに接続されていない場合は、ICカードセット案内画面へ画面遷移する。 ICカードから署名用認証局の自己署名証明書を取得する。 上位アプリケーションに署名用認証局の自己署名証明書を返す。 			

(2) 署名用電子証明書取得処理

項目名	署名用電子証明書取得処理			
概要	秘密鍵に対応する署名用電子証明書(DER形式)を取得			
	キー	型	値	内容
送信データ	command_type	int	0x01002002	コマンドタイプ
受信データ	command_type	int	0x01002002	コマンドタイプ
	p_cert	byte[]		秘密鍵に対応する署名用電子証明書(DER形式)
	err_code	int		エラーコード(4桁) 例外が発生した箇所のエラーコード

				の値が格納される。 正常系の場合は「0000」が格納される。
	detail_code	int		エラー詳細コード(8桁) 例外が発生した箇所の詳細コードの値が格納される。 正常系の場合は「00000000」が格納される。
備考	処理の流れは以下となる。 <ul style="list-style-type: none"> ・ ICカードに接続されていない場合は、ICカードセット案内画面へ画面遷移する。 ・ ICカードから秘密鍵に対応する署名用電子証明書を取得する。 ・ 上位アプリケーションに秘密鍵に対応する署名用電子証明書を返す。 			

(3) 署名生成処理(署名対象データを渡す場合)

項目名	署名生成処理(署名対象データを渡す場合)			
概要	署名対象データからハッシュ値を計算し、ICカードに格納された利用者秘密鍵を使用して署名値を生成			
	キー	型	値	内容
送信データ	command_type	int	0x01002003	コマンドタイプ
	message	byte[]		署名対象データ
	alg_id	int		ハッシュアルゴリズムの指定 SHA1:0 SHA256:1
受信データ	command_type	int	0x01002003	コマンドタイプ
	signature	byte[]		署名値
	err_code	int		エラーコード(4桁) 例外が発生した箇所のエラーコードの値が格納される。 正常系の場合は「0000」が格納される。
	detail_code	int		エラー詳細コード(8桁) 例外が発生した箇所の詳細コードの値が格納される。 正常系の場合は「00000000」が格納される。

備考	<p>処理の流れは以下となる。</p> <ul style="list-style-type: none"> IC カードに接続されていない場合は、IC カードセット案内画面へ画面遷移する。 署名対象データからハッシュ値を計算し、OID を付与し、IC カードに格納された利用者秘密鍵を使用して署名値を生成する。 上位アプリケーションに署名値を返す。 <p>ハッシュ値へのOID付与は以下のバイナリ形式で行う。</p> <ul style="list-style-type: none"> ハッシュアルゴリズムが SHA1 の場合(データサイズは 35Byte) <table border="1" style="width: 100%;"> <tr> <td>0x30, 0x21, 0x30, 0x09, 0x06, 0x05, 0x2b, 0x0e, 0x03, 0x02, 0x1a, 0x05, 0x00, 0x04, 0x14, SHA1 ハッシュ値(20Byte)</td> </tr> </table> ハッシュアルゴリズムが SHA256 の場合(データサイズは 51Byte) <table border="1" style="width: 100%;"> <tr> <td>0x30, 0x31, 0x30, 0x0d, 0x06, 0x09, 0x60, 0x86, 0x48, 0x01, 0x65, 0x03, 0x04, 0x02, 0x01, 0x05, 0x00, 0x04, 0x20, SHA256 ハッシュ値(32Byte)</td> </tr> </table> 	0x30, 0x21, 0x30, 0x09, 0x06, 0x05, 0x2b, 0x0e, 0x03, 0x02, 0x1a, 0x05, 0x00, 0x04, 0x14, SHA1 ハッシュ値(20Byte)	0x30, 0x31, 0x30, 0x0d, 0x06, 0x09, 0x60, 0x86, 0x48, 0x01, 0x65, 0x03, 0x04, 0x02, 0x01, 0x05, 0x00, 0x04, 0x20, SHA256 ハッシュ値(32Byte)
0x30, 0x21, 0x30, 0x09, 0x06, 0x05, 0x2b, 0x0e, 0x03, 0x02, 0x1a, 0x05, 0x00, 0x04, 0x14, SHA1 ハッシュ値(20Byte)			
0x30, 0x31, 0x30, 0x0d, 0x06, 0x09, 0x60, 0x86, 0x48, 0x01, 0x65, 0x03, 0x04, 0x02, 0x01, 0x05, 0x00, 0x04, 0x20, SHA256 ハッシュ値(32Byte)			

(4) 署名生成処理(ダイジェスト[ハッシュ値]を入力する場合)

項目名	署名生成処理(ダイジェスト[ハッシュ値]を入力する場合)			
概要	ハッシュ値に対して、IC カードに格納された利用者秘密鍵を使用して署名値を生成			
	キー	型	値	内容
送信データ	command_type	int	0x01002004	コマンドタイプ
	hash	byte[]		ダイジェスト(ハッシュ値)
	alg_id	int		ハッシュアルゴリズムの指定 SHA1:0 SHA256:1
受信データ	command_type	int	0x01002004	コマンドタイプ
	signature	byte[]		署名値
	err_code	int		エラーコード(4桁) 例外が発生した箇所のエラーコードの値が格納される。 正常系の場合は「0000」が格納される。
	detail_code	int		エラー詳細コード(8桁)

				例外が発生した箇所の詳細コードの値が格納される。 正常系の場合は「00000000」が格納される。
備考	<p>処理の流れは以下となる。</p> <ul style="list-style-type: none"> ICカードに接続されていない場合は、ICカードセット案内画面へ画面遷移する。 ハッシュ値に対して、ICカードに格納された利用者秘密鍵を使用して署名値を生成する。 上位アプリケーションに署名値を返す。 <p>ハッシュ値へのOID付与に関する注意事項は以下の通り。</p> <ul style="list-style-type: none"> 署名値生成の際、ダイジェスト（ハッシュ値）にOIDは付与しない。 OIDを付与したハッシュ値の署名値を生成する場合は、上位アプリケーションにてダイジェスト（ハッシュ値）にOIDを付与したデータを入力すること。（第5章 第2節 3（3）備考「ハッシュ値へのOID付与」を参照のこと） 			

（5）署名検証処理(検証対象データを渡す場合)

項目名	署名検証処理(検証対象データを渡す場合)			
概要	検証対象データからハッシュ値を計算し、ハッシュ値、署名値、公開鍵を使用して、署名値を検証			
	キー	型	値	内容
送信データ	command_type	int	0x01002005	コマンドタイプ
	certificate	byte[]		証明書データ(DER形式)
	message	byte[]		検証対象データ
	signature	byte[]		署名値
	alg_id	int		ハッシュアルゴリズムの指定 SHA1:0 SHA256:1
受信データ	command_type	int	0x01002005	コマンドタイプ
	verifyrec	int		署名値の検証結果 検証成功:0 検証失敗:-1
	err_code	int		エラーコード(4桁) 例外が発生した箇所のエラーコードの値が格納される。 正常系の場合は「0000」が格納される。

	detail_code	int		エラー詳細コード(8桁) 例外が発生した箇所の詳細コードの値が格納される。 正常系の場合は「00000000」が格納される。
備考	<p>処理の流れは以下となる。</p> <ul style="list-style-type: none"> ・ 検証対象データからハッシュ値を計算し、OIDを付与した値、署名値、公開鍵を使用して、署名値を検証する。 ・ 上位アプリケーションに検証結果を返す。 <p>署名値に関する注意事項は以下の通り。</p> <ul style="list-style-type: none"> ・ 上位アプリケーションはOIDを付与したハッシュ値から生成された署名値を入力すること。(第5章 第2節 3(3)備考「ハッシュ値へのOID付与」を参照のこと) 			

(6) 署名検証処理(ダイジェスト[ハッシュ値]を入力する場合)

項目名	署名検証処理(ダイジェスト[ハッシュ値]を入力する場合)			
概要	ハッシュ値、署名値、公開鍵を使用して、署名値を検証			
	キー	型	値	内容
送信データ	command_type	int	0x01002006	コマンドタイプ
	certificate	byte[]		証明書データ(DER形式)
	hash	byte[]		ダイジェストデータ
	signature	byte[]		署名値
	alg_id	int		ハッシュアルゴリズムの指定 SHA1:0 SHA256:1
受信データ	command_type	int	0x01002006	コマンドタイプ
	verifyrec	int		署名値の検証結果 検証成功:0 検証失敗:-1
	err_code	int		エラーコード(4桁) 例外が発生した箇所のエラーコードの値が格納される。 正常系の場合は「0000」が格納される。
	detail_code	int		エラー詳細コード(8桁) 例外が発生した箇所の詳細コードの値が格納される。 正常系の場合は「00000000」が格

				納される。													
備考	<p>処理の流れは以下となる。</p> <ul style="list-style-type: none"> ハッシュ値、署名値、公開鍵を使用して、署名値を検証する。 上位アプリケーションに検証結果を返す。 <p>ダイジェストデータと署名値に関する注意事項は以下の通り。</p> <ul style="list-style-type: none"> OID を付与したハッシュ値から生成された署名値を検証する場合、上位アプリケーションにてダイジェストデータにOID を付与したデータを入力すること。(第5章 第2節 3(3)備考「ハッシュ値へのOID付与」を参照のこと) OID を付与していないハッシュ値から生成された署名値を検証する場合、上位アプリケーションにてダイジェストデータにOID を付与していないデータを入力すること。 <table border="1"> <thead> <tr> <th colspan="2" rowspan="2"></th> <th colspan="2">ダイジェストデータ</th> </tr> <tr> <th>OID あり</th> <th>OID なし</th> </tr> </thead> <tbody> <tr> <th rowspan="2">署名値</th> <th>OID あり</th> <td>OK</td> <td>NG</td> </tr> <tr> <th>OID なし</th> <td>NG</td> <td>OK</td> </tr> </tbody> </table>						ダイジェストデータ		OID あり	OID なし	署名値	OID あり	OK	NG	OID なし	NG	OK
		ダイジェストデータ															
		OID あり	OID なし														
署名値	OID あり	OK	NG														
	OID なし	NG	OK														

4 カード AP ライブラリ(利用者証明用の場合)

(1) 利用者証明用認証局の自己署名証明書取得処理

項目名	利用者証明用認証局の自己署名証明書取得処理			
概要	利用者証明用認証局の自己署名証明書(DER形式)を取得			
	キー	型	値	内容
送信データ	command_type	int	0x01001001	コマンドタイプ
受信データ	command_type	int	0x01001001	コマンドタイプ
	ca_cert	byte[]		利用者証明用認証局の自己署名証明書(DER形式)
	err_code	int		エラーコード(4桁) 例外が発生した箇所のエラーコードの値が格納される。 正常系の場合は「0000」が格納される。
	detail_code	int		エラー詳細コード(8桁) 例外が発生した箇所の詳細コードの値が格納される。 正常系の場合は「00000000」が格納される。
備考	<p>処理の流れは以下となる。</p> <ul style="list-style-type: none"> ICカードに接続されていない場合は、ICカードセット案内画面へ画面遷移する。 ICカードから利用者証明用認証局の自己署名証明書を取得する。 上位アプリケーションに利用者証明用認証局の自己署名証明書を返す。 			

(2) 利用者証明用電子証明書取得処理

項目名	利用者証明用電子証明書取得処理			
概要	秘密鍵に対応する利用者証明用電子証明書(DER形式)を取得			
	キー	型	値	内容
送信データ	command_type	int	0x01001002	コマンドタイプ
受信データ	command_type	int	0x01001002	コマンドタイプ
	p_cert	byte[]		秘密鍵に対応する利用者証明用電子証明書(DER形式)
	err_code	int		エラーコード(4桁) 例外が発生した箇所のエラーコ

				ードの値が格納される。 正常系の場合は「0000」が格納される。
	detail_code	int		エラー詳細コード(8桁) 例外が発生した箇所の詳細コードの値が格納される。 正常系の場合は「00000000」が格納される。
備考	処理の流れは以下となる。 <ul style="list-style-type: none"> ・ ICカードに接続されていない場合は、ICカードセット案内画面へ画面遷移する。 ・ ICカードから秘密鍵に対応する利用者証明用電子証明書を取得する。 ・ 上位アプリケーションに秘密鍵に対応する利用者証明用電子証明書を返す。 			

(3) 署名生成処理 (署名対象データを渡す場合)

項目名	署名生成処理 (署名対象データを渡す場合)			
概要	署名対象データからハッシュ値を計算し、ICカードに格納された利用者秘密鍵を使用して署名値を生成			
	キー	型	値	内容
送信データ	command_type	int	0x01001003	コマンドタイプ
	message	byte[]		署名対象データ
	alg_id	int		ハッシュアルゴリズムの指定 SHA1:0 SHA256:1
受信データ	command_type	int	0x01001003	コマンドタイプ
	signature	byte[]		署名値
	err_code	int		エラーコード(4桁) 例外が発生した箇所のエラーコードの値が格納される。 正常系の場合は「0000」が格納される。
	detail_code	int		エラー詳細コード(8桁) 例外が発生した箇所の詳細コードの値が格納される。 正常系の場合は「00000000」が格納される。
備考	処理の流れは以下となる。 <ul style="list-style-type: none"> ・ ICカードに接続されていない場合は、ICカードセット案内画面へ画面遷 			

	<p>移す。</p> <ul style="list-style-type: none"> 署名対象データからハッシュ値を計算し、OIDを付与し、ICカードに格納された利用者秘密鍵を使用して署名値を生成する。 上位アプリケーションに署名値を返す。 <p>ハッシュ値へのOID付与は以下のバイナリ形式で行う。</p> <ul style="list-style-type: none"> 第5章 第2節 3(3)備考「ハッシュ値へのOID付与」を参照のこと。
--	---

(4) 署名生成処理(ダイジェスト[ハッシュ値]を入力する場合)

項目名	署名生成処理(ダイジェスト[ハッシュ値]を入力する場合)			
概要	ハッシュ値に対して、ICカードに格納された利用者秘密鍵を使用して署名値を生成			
	キー	型	値	内容
送信データ	command_type	int	0x01001004	コマンドタイプ
	hash	byte[]		ダイジェスト(ハッシュ値)
	alg_id	int		ハッシュアルゴリズムの指定 SHA1:0 SHA256:1
受信データ	command_type	int	0x01001004	コマンドタイプ
	signature	byte[]		署名値
	err_code	int		エラーコード(4桁) 例外が発生した箇所のエラーコードの値が格納される。 正常系の場合は「0000」が格納される。
	detail_code	int		エラー詳細コード(8桁) 例外が発生した箇所の詳細コードの値が格納される。 正常系の場合は「00000000」が格納される。
備考	<p>処理の流れは以下となる。</p> <ul style="list-style-type: none"> ICカードに接続されていない場合は、ICカードセット案内画面へ画面遷移する。 ハッシュ値に対して、ICカードに格納された利用者秘密鍵を使用して署名値を生成する。 上位アプリケーションに署名値を返す。 <p>ハッシュ値へのOID付与に関する注意事項は以下の通り。</p>			

- | | |
|--|---|
| | <ul style="list-style-type: none">・ 署名値生成の際、ダイジェスト（ハッシュ値）に OID は付与しない。・ OID を付与したハッシュ値の署名値を生成する場合は、上位アプリケーションにてダイジェスト（ハッシュ値）に OID を付与したデータを入力すること。（第 5 章 第 2 節 3（3）備考「ハッシュ値への OID 付与」を参照のこと） |
|--|---|

（5）署名検証処理（検証対象データを渡す場合）

「第 5 章 第 2 節 3（5）」を参照。

（6）署名検証処理（ダイジェスト [ハッシュ値] を入力する場合）

「第 5 章 第 2 節 3（6）署名検証処理（ダイジェスト [ハッシュ値] を入力する場合）」を参照。

5 個人認証サービス AP(署名用、利用者証明用共通)

(1) 証明書表示

項目名	証明書表示			
概要	電子証明書を表示			
	キー	型	値	内容
送信データ	command_type	int	0x01003001	コマンドタイプ
	cert	byte[]		証明書データ(DER形式)
受信データ	command_type	int	0x01003001	コマンドタイプ
	err_code	int		エラーコード(4桁) 例外が発生した箇所のエラーコードの値が格納される。 正常系の場合は「0000」が格納される。
	detail_code	int		エラー詳細コード(8桁) 例外が発生した箇所の詳細コードの値が格納される。 正常系の場合は「00000000」が格納される。
備考	処理の流れは以下となる。 <ul style="list-style-type: none"> 電子証明書表示画面へ画面遷移する。 電子証明書表示画面で証明書データの内容を表示する。 			

(2) 基本4情報取得

項目名	基本4情報取得			
概要	署名用電子証明書から基本4情報(氏名、住所、性別、生年月日)を取得			
	キー	型	値	内容
送信データ	command_type	int	0x01003002	コマンドタイプ
	cert	byte[]		証明書データ(DER形式)
受信データ	command_type	int	0x01003002	コマンドタイプ
	address	String		住所
	date_of_birth	String		生年月日 (第6章 第11節 設定ルール一覧参照)
	gender	String		性別 (第6章 第11節 設定ルール一覧参照)

	name	String		氏名 (大括弧「 [] 」内は旧氏を示す。括弧「 () 」内は通称を示す。)
	substitute_character_of_address	String		住所の代替文字 (第6章 第11節 設定ルール一覧参照)
	substitute_character_of_name	String		氏名の代替文字 (第6章 第11節 設定ルール一覧)
	err_code	int		エラーコード(4桁) 例外が発生した箇所のエラーコードの値が格納される。 正常系の場合は「0000」が格納される。
	detail_code	int		エラー詳細コード(8桁) 例外が発生した箇所の詳細コードの値が格納される。 正常系の場合は「00000000」が格納される。
備考	処理の流れは以下となる。 <ul style="list-style-type: none"> ・ 証明書データから基本4情報を取得する。 ・ 上位アプリケーションに基本4情報を返す。 			

(3) 官職証明書検証

項目名	官職証明書検証			
概要	官職証明書や職責証明書の証明書検証を行うため、公的個人認証サービスの官職証明書検証サービスに対して証明書検証要求を発行			
	キー	型	値	内容
送信データ	command_type	int	0x01003003	コマンドタイプ
	cert	byte[]		証明書データ(DER形式)
受信データ	command_type	int	0x01003003	コマンドタイプ
	cert_status	int		官職証明書検証機能で取得した証明書の検証結果コード (表 5-4 表-5-4 検証結果コードの値一覧 参照)
	cert_path_status	int		官職証明書検証機能で取得した認証パス検証結果コード

			(表 5-5表-5-5認証パス 検証結果コードの値一覧認証 パス検証結果コードの値一覧 参照)
	response_status	int	官職証明書検証機能で取得し た有効性検証結果コード (表 5-6表-5-6有効性検 証結果コードの値一覧有効性 検証結果コードの値一覧参 照)
	err_code	int	エラーコード(4桁) 例外が発生した箇所のエラー コードの値が格納される。 正常系の場合は「0000」が格 納される。
	detail_code	int	エラー詳細コード(8桁) 例外が発生した箇所の詳細コ ードの値が格納される。 正常系の場合は「00000000」 が格納される。
備考	<p>処理の流れは以下となる。</p> <ul style="list-style-type: none"> ・ ICカードに接続されていない場合は、ICカードセット案内画面へ画面遷移する。 ・ 官職証明書検証を行い、結果を取得する。 ・ 上位アプリケーションに結果を返す。 		

表 5-4 検証結果コードの値一覧

値	意味
0	検証成功。
-1	認証パスの構築および検証に失敗。
-8	有効性検証に失敗。

表 5-5 認証パス検証結果コードの値一覧

値	意味
0	認証パスの構築が成功し検証結果が正しい。
101	認証パス構築ができない。
202	認証パスに署名が不正である証明書が含まれる。

値	意味
203	認証パスに失効した証明書が含まれる。
204	認証パスにポリシーが一致しない証明書が含まれる。
205	認証パスに制約に違反している証明書が含まれる。
301	証明書検証サーバ(CVS)側で検証対象証明書の受け付けを拒否した。
302	証明書検証サーバ(CVS)側で中間証明書または TA 証明書の受け付けを拒否した。
901	証明書検証サーバ(CVS)側で要求の受け付けを拒否した。
902	要求がタイムアウトとなった。

表 5 - 6 有効性検証結果コードの値一覧

値	意味
0	OCSP Request が正しく処理された。
1	OCSP Request のフォーマットエラー。
2	内部エラー。
3	一時的な回答不能。
5	OCSP Request への署名が必要。
6	クライアントが認証されていない。

(4) 自分の電子証明書の有効性確認

項目名	自分の電子証明書の有効性確認			
概要	IC カード内の自分の電子証明書(利用者証明書)の有効性を確認するために、公的個人認証サービスのオンライン窓口サービスに対して有効性確認要求を発行			
	キー	型	値	内容
送信データ	command_type	int	0x01003004	コマンドタイプ
	cert	byte[]		証明書データ(DER 形式)
受信データ	command_type	int	0x01003004	コマンドタイプ
	code	int		有効性確認結果 (表 5 - 7表 5 - 7有効性確認結果一覧有効性確認結果一覧参照)
	exception	String		有効性確認中に発生した Exception のメッセージ
	confirm_msg	String		確認結果メッセージ
	err_code	int		エラーコード(4 桁) 例外が発生した箇所のエラーコードの値が格納される。

				正常系の場合は「0000」が格納される。
	detail_code	int		エラー詳細コード(8桁) 例外が発生した箇所の詳細コードの値が格納される。 正常系の場合は「00000000」が格納される。
備考	<p>処理の流れは以下となる。</p> <ul style="list-style-type: none"> ・ ICカードに接続されていない場合は、ICカードセット案内画面へ画面遷移する。 ・ 有効性確認を行い、結果を取得する。 ・ 上位アプリケーションに結果を返す。 			

表 5-7 有効性確認結果一覧

値	意味
200	有効性確認結果が有効である。
600	申請書の電子署名検証エラーが発生した。
601	オンライン窓口サーバで「申請情報取込エラー」が発生した。
602	オンライン窓口サーバで「対象とする証明書の有効期間切れエラー」が発生した。
603	オンライン窓口サーバで「受信データ形式エラー」が発生した。
604	オンライン窓口サーバで「対象とする証明書の認証局の電子署名検証エラー」が発生した。
605	オンライン窓口サーバで「対象とする証明書の有効性確認エラー」が発生した。
606	オンライン窓口サーバで「対象とする証明書のオンライン窓口エラー(OCSP)」が発生した。
607	対象とする証明書の失効済である。
608	対象とする証明書が失効申請済みである。(二重申請エラー/既オンライン失効申請エラー)
609	オンライン窓口サーバで「オンライン窓口エラー(DB)」が発生した。
611	オンライン窓口サーバが混雑している。
612	オンライン窓口でエラーが発生した(コードなし)。
613	対象とする証明書の一時保留状態である。
622	オンライン窓口サーバで「対象とする証明書と署名実施証明書の不一致エラー」が発生した。
700	ログインのキャンセル。
701	メモリ不足が発生した。

値	意味
702	予期せぬエラーが発生した。
703	ICカードがロックされている。
704	ICカードに接続できない。
705	対象とする証明書の解析に失敗。
706	サーバ時刻の取得に失敗。
707	オンライン窓口サーバへのアクセスに失敗。
708	対象とする証明書の有効期間切れ。
709	対象とする証明書の有効期間開始の日付が未来の日時。
100 ~ 500 (200を除く)	HTTP1.1 ステータスコード(100 ~ 500 番台)。

(5) ICカード種別取得

項目名	ICカード種別取得			
概要	セットされているICカードの種別を取得			
	キー	型	値	内容
送信データ	command_type	int	0x01003005	コマンドタイプ
受信データ	command_type	int	0x01003005	コマンドタイプ
	id	int		ICカード種別 (表 5-8 表 5-8-ICカード種別一覧 ICカード種別一覧参照)
	token	String		ICカード token 情報
	err_code	int		エラーコード(4桁) 例外が発生した箇所のエラーコードの値が格納される。 正常系の場合は「0000」が格納される。
	detail_code	int		エラー詳細コード(8桁) 例外が発生した箇所の詳細コードの値が格納される。 正常系の場合は「00000000」が格納される。
備考	<p>処理の流れは以下となる。</p> <ul style="list-style-type: none"> ICカードに接続されていない場合は、ICカードセット案内画面へ画面遷移する。 ICカードにアクセスしてICカードの種別を取得する。 上位アプリケーションに結果を返す。 			

表 5 - 8 IC カード種別一覧

値	意味
0	IC カード種別の不明なカード
1	IC カード種別の住基カード
2	IC カード種別の個人番号カード

第 3 節 コーリングシーケンス

以下のシーケンスは上位アプリケーション内での処理となる。

1 初期処理

Intent インスタンスの生成	引数 ・ 第一引数: Intent.ACTION_SEND
intent.setClassName	JPKI 利用者ソフトのパッケージ名とクラス名の設定 引数 ・ 第一引数: "jp.go.jpki.mobile.utility" ・ 第二引数: "jp.go.jpki.mobile.intent.JPKIIntentActivity"
intent.putExtra	拡張データの設定(コマンドタイプの設定) 引数 ・ 第一引数: "command_type" ・ 第二引数: 0x01004001
startActivityForResult	Android 版 JPKI 利用者ソフトとのインテント開始処理 引数 ・ 第一引数: intent ・ 第二引数: 任意のリクエストコード(int 型)
Android 版 JPKI 利用者ソフトからの情報を onActivityResult(int requestCode, int resultCode, Intent data)で受け取る	
data.getIntExtra	コマンドタイプを取得 引数 ・ 第一引数: "command_type"
data.getBooleanExtra	結果を取得(boolean 型) 引数 ・ 第一引数: "result" 取得した値が false の場合は送信開始処理失敗
data.getIntExtra	エラーコードを取得(int 型) 引数 ・ 第一引数: "err_code" 取得した値が 0000 の場合は正常に処理された

data.getIntExtra	詳細コードを取得(int 型) 引数 ・ 第一引数: “ detail_code ” 取得した値が 00000000 の場合は正常に処理された
------------------	--

2 終了処理

Intent インスタンスの生成	引数 ・ 第一引数: Intent.ACTION_SEND
intent.setClassName	JPKI 利用者ソフトのパッケージ名とクラス名の設定 引数 ・ 第一引数: “ jp.go.jpki.mobile.utility ” ・ 第二引数: “ jp.go.jpki.mobile.intent.JPKIIntentActivity ”
intent.putExtra	拡張データの設定(コマンドタイプの設定) 引数 ・ 第一引数: “ command_type ” ・ 第二引数: 0x01004002
startActivityForResult	Android 版 JPKI 利用者ソフトとのインテント開始処理 引数 ・ 第一引数: intent ・ 第二引数: 任意のリクエストコード(int 型)
Android 版 JPKI 利用者ソフトからの情報を onActivityResult(int requestCode, int resultCode, Intent data) で受け取る	
data.getIntExtra	コマンドタイプを取得 引数 ・ 第一引数: “ command_type ”
data.getBooleanExtra	結果を取得(boolean 型) 引数 ・ 第一引数: “ result ” 取得した値が false の場合は送信終了処理失敗
data.getIntExtra	エラーコードを取得(int 型) 引数 ・ 第一引数: “ err_code ” 取得した値が 0000 の場合は正常に処理された
data.getIntExtra	詳細コードを取得(int 型)

	<p>引数</p> <ul style="list-style-type: none">・ 第一引数: “detail_code” <p>取得した値が 00000000 の場合は正常に処理された</p>
--	---

3 カード AP ライブラリ(署名用の場合)

(1) 署名用認証局の自己署名証明書取得処理

初期処理	1 初期処理 初期処理 参照
Intent インスタンスの生成	引数 ・ 第一引数: Intent.ACTION_SEND
intent.setClassName	JPKI 利用者ソフトのパッケージ名とクラス名の設定 引数 ・ 第一引数: “jp.go.jpki.mobile.utility” ・ 第二引数: “jp.go.jpki.mobile.intent.JPKIIntentActivity”
intent.putExtra	拡張データの設定(コマンドタイプの設定) 引数 ・ 第一引数: “command_type” ・ 第二引数: 0x01002001
startActivityForResult	Android 版 JPKI 利用者ソフトの起動と intent 送信 引数 ・ 第一引数: intent ・ 第二引数: 任意のリクエストコード(int 型)
Android 版 JPKI 利用者ソフトからの情報を onActivityResult(int requestCode, int resultCode, Intent data)で受け取る	
data.getIntExtra	コマンドタイプを取得 引数 ・ 第一引数: “command_type”
data.getByteArrayExtra	署名用認証局の自己署名証明書(DER 形式)の取得 引数 ・ 第一引数: “ca_cert”
data.getIntExtra	エラーコードを取得(int 型) 引数 ・ 第一引数: “err_code” 取得した値が 0000 の場合は正常に処理された

data.getIntExtra	詳細コードを取得(int型) 引数 ・第一引数: “detail_code” 取得した値が00000000の場合は正常に処理された
------------------	--

終了処理	2 終了処理 終了処理 参照
------	--

(2) 署名用電子証明書取得処理

初期処理	1 初期処理 初期処理 参照
------	--

Intent インスタンスの生成	引数 ・第一引数: Intent.ACTION_SEND
------------------	---------------------------------

intent.setClassName	JPKI 利用者ソフトのパッケージ名とクラス名の設定 引数 ・第一引数: “jp.go.jpki.mobile.utility” ・第二引数: “jp.go.jpki.mobile.intent.JPKIIntentActivity”
---------------------	---

intent.putExtra	拡張データの設定(コマンドタイプの設定) 引数 ・第一引数: “command_type” ・第二引数: 0x01002002
-----------------	--

startActivityForResult	Android 版 JPKI 利用者ソフトの起動と intent 送信 引数 ・第一引数: intent ・第二引数: 任意のリクエストコード(int型)
------------------------	--

Android 版 JPKI 利用者ソフトからの情報を onActivityResult(int requestCode, int resultCode, Intent data)で受け取る	
---	--

data.getIntExtra	コマンドタイプを取得 引数 ・第一引数: “command_type”
------------------	---

data.getByteArrayExtra	秘密鍵に対応する利用者証明書(DER形式)の取得
------------------------	--------------------------

	引数 ・ 第一引数: “ p_cert ”
--	--------------------------

data.getIntExtra	エラーコードを取得(int 型) 引数 ・ 第一引数: “ err_code ” 取得した値が 0000 の場合は正常に処理された
------------------	--

data.getIntExtra	詳細コードを取得(int 型) 引数 ・ 第一引数: “ detail_code ” 取得した値が 00000000 の場合は正常に処理された
------------------	--

終了処理	2 終了処理終了処理参照
------	------------------------------

(3) 署名生成処理 (署名対象データを渡す場合)

初期処理	1 初期処理初期処理参照
------	------------------------------

Intent インスタンスの生成	引数 ・ 第一引数: Intent.ACTION_SEND
------------------	----------------------------------

intent.setClassName	JPKI 利用者ソフトのパッケージ名とクラス名の設定 引数 ・ 第一引数: “ jp.go.jpki.mobile.utility ” ・ 第二引数: “ jp.go.jpki.mobile.intent.JPKIIntentActivity ”
---------------------	---

intent.putExtra	拡張データの設定(コマンドタイプの設定) 引数 ・ 第一引数: “ command_type ” ・ 第二引数: 0x01002003
-----------------	--

intent.putExtra	拡張データの設定(署名対象データの設定) (1),(2)で取得できるデータ 引数 ・ 第一引数: “ message ” ・ 第二引数: byte[]型
-----------------	--

intent.putExtra	拡張データの設定(ハッシュアルゴリズムの指定)
-----------------	-------------------------

	引数 <ul style="list-style-type: none"> ・ 第一引数: “alg_id” ・ 第二引数: int 型
--	---

startActivityForResult	Android 版 JPKI 利用者ソフトの起動と intent 送信 引数 <ul style="list-style-type: none"> ・ 第一引数: intent ・ 第二引数: 任意のリクエストコード(int 型)
------------------------	---

Android 版 JPKI 利用者ソフトからの情報を onActivityResult(int requestCode, int resultCode, Intent data)で受け取る

data.getIntExtra	コマンドタイプを取得 引数 <ul style="list-style-type: none"> ・ 第一引数: “command_type”
------------------	--

data.getByteArrayExtra	署名値の取得 引数 <ul style="list-style-type: none"> ・ 第一引数: “signature”
------------------------	---

data.getIntExtra	エラーコードを取得(int 型) 引数 <ul style="list-style-type: none"> ・ 第一引数: “err_code” 取得した値が 0000 の場合は正常に処理された
------------------	---

data.getIntExtra	詳細コードを取得(int 型) 引数 <ul style="list-style-type: none"> ・ 第一引数: “detail_code” 取得した値が 00000000 の場合は正常に処理された
------------------	---

終了処理	2 終了処理 終了処理 参照
------	--

(4) 署名生成処理 (ダイジェスト [ハッシュ値] を入力する場合)

初期処理	1 初期処理 初期処理 参照
------	--

Intent インスタンスの生成	引数 第一引数: Intent.ACTION_SEND
------------------	--------------------------------

intent.setClassName	JPKI 利用者ソフトのパッケージ名とクラス名の設定
---------------------	----------------------------

	引数 <ul style="list-style-type: none"> ・ 第一引数: “ jp.go.jpki.mobile.utility ” ・ 第二引数: “ jp.go.jpki.mobile.intent.JPKIIntentActivity ”
intent.putExtra	拡張データの設定(コマンドタイプの設定) 引数 <ul style="list-style-type: none"> ・ 第一引数: “ command_type ” ・ 第二引数: 0x01002004
intent.putExtra	拡張データの設定(ダイジェスト(ハッシュ値)の設定) 引数 <ul style="list-style-type: none"> ・ 第一引数: “ hash ” ・ 第二引数: byte[]型
intent.putExtra	拡張データの設定(ハッシュアルゴリズムの指定) 引数 <ul style="list-style-type: none"> ・ 第一引数: “ alg_id ” ・ 第二引数: int 型
startActivityForResult	Android 版 JPKI 利用者ソフトの起動と intent 送信 引数 <ul style="list-style-type: none"> ・ 第一引数: intent ・ 第二引数: 任意のリクエストコード(int 型)
Android 版 JPKI 利用者ソフトからの情報を onActivityResult(int requestCode, int resultCode, Intent data)で受け取る	
data.getIntExtra	コマンドタイプを取得 引数 <ul style="list-style-type: none"> ・ 第一引数: “ command_type ”
data.getByteArrayExtra	署名値の取得 引数 <ul style="list-style-type: none"> ・ 第一引数: “ signature ”
data.getIntExtra	エラーコードを取得(int 型) 引数 <ul style="list-style-type: none"> ・ 第一引数: “ err_code ”

	取得した値が 0000 の場合は正常に処理された
--	--------------------------

data.getIntExtra	詳細コードを取得(int 型) 引数 ・ 第一引数: “ detail_code ” 取得した値が 00000000 の場合は正常に処理された
------------------	--

終了処理	2 終了処理終了処理参照
------	------------------------------

(5) 署名検証処理(検証対象データを渡す場合)

初期処理	1 初期処理初期処理参照
------	------------------------------

Intent インスタンスの生成	引数 ・ 第一引数: Intent.ACTION_SEND
------------------	----------------------------------

intent.setClassName	JPKI 利用者ソフトのパッケージ名とクラス名の設定 引数 ・ 第一引数: “ jp.go.jpki.mobile.utility ” ・ 第二引数: “ jp.go.jpki.mobile.intent.JPKIIntentActivity ”
---------------------	---

intent.putExtra	拡張データの設定(コマンドタイプの設定) 引数 ・ 第一引数: “ command_type ” ・ 第二引数: 0x01002005
-----------------	--

intent.putExtra	拡張データの設定(証明書データ(DER 形式)) (1), (2)で取得できるデータ 引数 ・ 第一引数: “ certificate ” ・ 第二引数: byte[]型
-----------------	---

intent.putExtra	拡張データの設定(検証対象データの設定) 引数 ・ 第一引数: “ message ” ・ 第二引数: byte[]型
-----------------	--

intent.putExtra	拡張データの設定(署名値の設定) (3), (4)で取得できるデータ
-----------------	---------------------------------------

	引数 ・ 第一引数: “signature” ・ 第二引数: byte[]型
--	--

intent.putExtra	拡張データの設定(ハッシュアルゴリズムの指定) 引数 ・ 第一引数: “alg_id” ・ 第二引数: int 型
-----------------	--

startActivityForResult	Android 版 JPKI 利用者ソフトの起動と intent 送信 引数 ・ 第一引数: intent ・ 第二引数: 任意のリクエストコード(int 型)
------------------------	---

Android 版 JPKI 利用者ソフトからの情報を onActivityResult(int requestCode, int resultCode, Intent data)で受け取る

data.getIntExtra	コマンドタイプを取得 引数 ・ 第一引数: “command_type”
------------------	--

data.getIntExtra	署名値の検証結果 引数 ・ 第一引数: “verifyrec”
------------------	---------------------------------------

data.getIntExtra	エラーコードを取得(int 型) 引数 ・ 第一引数: “err_code” 取得した値が 0000 の場合は正常に処理された
------------------	--

data.getIntExtra	詳細コードを取得(int 型) 引数 ・ 第一引数: “detail_code” 取得した値が 00000000 の場合は正常に処理された
------------------	--

終了処理

2 [終了処理終了処理参照](#)

(6) 署名検証処理 (ダイジェスト [ハッシュ値] を入力する場合)

初期処理

1 [初期処理初期処理参照](#)

Intent インスタンスの生成	引数 ・ 第一引数: Intent.ACTION_SEND
intent.setClassName	JPKI 利用者ソフトのパッケージ名とクラス名の設定 引数 ・ 第一引数: “jp.go.jpki.mobile.utility” ・ 第二引数: “jp.go.jpki.mobile.intent.JPKIIntentActivity”
intent.putExtra	拡張データの設定(コマンドタイプの設定) 引数 ・ 第一引数: “command_type” ・ 第二引数: 0x01002006
intent.putExtra	拡張データの設定(証明書データ(DER 形式)) (1),(2)で取得できるデータ 引数 ・ 第一引数: “certificate” ・ 第二引数: byte[]型
intent.putExtra	拡張データの設定(ダイジェストデータの設定) 引数 ・ 第一引数: “hash” ・ 第二引数: byte[]型
intent.putExtra	拡張データの設定(署名値の設定) (3),(4)で取得できるデータ 引数 ・ 第一引数: “signature” ・ 第二引数: byte[]型
intent.putExtra	拡張データの設定(ハッシュアルゴリズムの指定) 引数 ・ 第一引数: “alg_id” ・ 第二引数: int 型
startActivityForResult	Android 版 JPKI 利用者ソフトの起動と intent 送信 引数 ・ 第一引数: intent

	・第二引数: 任意のリクエストコード(int 型)
Android 版 JPKI 利用者ソフトからの情報を onActivityResult(int requestCode, int resultCode, Intent data)で受け取る	
data.getIntExtra	コマンドタイプを取得 引数 ・第一引数: “ command_type ”
data.getIntExtra	署名値の検証結果 引数 ・第一引数: “ verifyrec ”
data.getIntExtra	エラーコードを取得(int 型) 引数 ・第一引数: “ err_code ” 取得した値が 0000 の場合は正常に処理された
data.getIntExtra	詳細コードを取得(int 型) 引数 ・第一引数: “ detail_code ” 取得した値が 00000000 の場合は正常に処理された
終了処理	2 終了処理終了処理参照

(7) 繰り返し署名生成処理(署名対象データを渡すパターン)

繰り返し生成処理に関しては、「(3) [署名生成処理 \(署名対象データを渡す場合\)](#) [署名生成処理 \(署名対象データを渡す場合\)](#)」網掛け部分を署名対象データの個数分だけ繰り返して呼び出す。

(8) 繰り返し署名生成処理(ダイジェスト[ハッシュ値]を渡すパターン)

繰り返し生成処理に関しては、「(4) [署名生成処理 \(ダイジェスト\[ハッシュ値\]を入力する場合\)](#) [署名生成処理 \(ダイジェスト\[ハッシュ値\]を入力する場合\)](#)」網掛け部分を署名対象データの個数分だけ繰り返して呼び出す。

(9) 繰り返し署名検証処理(検証対象データを渡すパターン)

「(5) [署名検証処理 \(検証対象データを渡す場合\)](#) [署名検証処理 \(検証対象データを渡す場合\)](#)」の網掛け部分を検証対象データの個数分だけ繰り返して呼び出す。(但し、すべての電子署名が同一の秘密鍵で生成された場合とする。)

(1 0) 繰り返し署名検証処理(ハッシュ値を渡すパターン)

「(6) 署名検証処理 (ダイジェスト [ハッシュ値] を入力する場合)」の網掛け部分を検証対象データの個数分だけ繰り返して呼び出す。(但し、すべての電子署名が同一の秘密鍵で生成された場合とする。)

4 カード AP ライブラリ(利用者証明用の場合)

(1) 利用者証明用認証局の自己署名証明書取得処理

初期処理	1 初期処理 初期処理 参照
Intent インスタンスの生成	引数 ・ 第一引数: Intent.ACTION_SEND
intent.setClassName	JPKI 利用者ソフトのパッケージ名とクラス名の設定 引数 ・ 第一引数: “jp.go.jpki.mobile.utility” ・ 第二引数: “jp.go.jpki.mobile.intent.JPKIIntentActivity”
intent.putExtra	拡張データの設定(コマンドタイプの設定) 引数 ・ 第一引数: “command_type” ・ 第二引数: 0x01001001
startActivityForResult	Android 版 JPKI 利用者ソフトの起動と intent 送信 引数 ・ 第一引数: intent ・ 第二引数: 任意のリクエストコード(int 型)
Android 版 JPKI 利用者ソフトからの情報を onActivityResult(int requestCode, int resultCode, Intent data)で受け取る	
data.getIntExtra	コマンドタイプを取得 引数 ・ 第一引数: “command_type”
data.getByteArrayExtra	利用者証明用認証局の自己署名証明書(DER 形式)の取得 引数 ・ 第一引数: “ca_cert”
data.getIntExtra	エラーコードを取得(int 型) 引数 ・ 第一引数: “err_code” 取得した値が 0000 の場合は正常に処理された

data.getIntExtra	<p>詳細コードを取得(int 型)</p> <p>引数</p> <ul style="list-style-type: none"> ・ 第一引数: “ detail_code ” <p>取得した値が 00000000 の場合は正常に処理された</p>
------------------	--

終了処理	2 終了処理終了処理参照
------	------------------------------

(2) 利用者証明用電子証明書取得処理

初期処理	1 初期処理初期処理参照
------	------------------------------

Intent インスタンスの生成	<p>引数</p> <ul style="list-style-type: none"> ・ 第一引数: Intent.ACTION_SEND
------------------	--

intent.setClassName	<p>インテントの送信元のアプリケーションコンテキストと対象アプリの設定</p> <p>引数</p> <ul style="list-style-type: none"> ・ 第一引数: “ jp.go.jpki.mobile.utility ” ・ 第二引数: “ jp.go.jpki.mobile.intent.JPKIIntentActivity ”
---------------------	---

intent.putExtra	<p>拡張データの設定(コマンドタイプの設定)</p> <p>引数</p> <ul style="list-style-type: none"> ・ 第一引数: “ command_type ” ・ 第二引数: 0x01001002
-----------------	--

startActivityForResult	<p>Android 版 JPKI 利用者ソフトの起動と intent 送信</p> <p>引数</p> <ul style="list-style-type: none"> ・ 第一引数: intent ・ 第二引数: 任意のリクエストコード(int 型)
------------------------	---

Android 版 JPKI 利用者ソフトからの情報を onActivityResult(int requestCode, int resultCode, Intent data)で受け取る	
---	--

data.getIntExtra	<p>コマンドタイプを取得</p> <p>引数</p> <ul style="list-style-type: none"> ・ 第一引数: “ command_type ”
------------------	--

data.getByteArrayExtra	秘密鍵に対応する利用者証明書(DER 形式)の取得
------------------------	---------------------------

	引数 ・第一引数: “ p_cert ”
--	-------------------------

data.getIntExtra	エラーコードを取得(int 型) 引数 ・第一引数: “ err_code ” 取得した値が 0000 の場合は正常に処理された
------------------	---

data.getIntExtra	詳細コードを取得(int 型) 引数 ・第一引数: “ detail_code ” 取得した値が 00000000 の場合は正常に処理された
------------------	---

終了処理	2 終了処理終了処理参照
------	------------------------------

(3) 署名生成処理 (署名対象データを渡す場合)

初期処理	1 初期処理初期処理参照
------	------------------------------

Intent インスタンスの生成	引数 ・第一引数: Intent.ACTION_SEND
------------------	---------------------------------

intent.setClassName	JPKI 利用者ソフトのパッケージ名とクラス名の設定 引数 ・第一引数: “ jp.go.jpki.mobile.utility ” ・第二引数: “ jp.go.jpki.mobile.intent.JPKIIntentActivity ”
---------------------	---

intent.putExtra	拡張データの設定(コマンドタイプの設定) 引数 ・第一引数: “ command_type ” ・第二引数: 0x01001003
-----------------	--

intent.putExtra	拡張データの設定(署名対象データの設定) (1),(2)で取得できるデータ 引数 ・第一引数: “ message ” ・第二引数: byte[]型
-----------------	--

intent.putExtra	拡張データの設定(ハッシュアルゴリズムの指定)
-----------------	-------------------------

	引数 ・ 第一引数: “alg_id” ・ 第二引数: int 型
--	---

startActivityForResult	Android 版 JPKI 利用者ソフトの起動と intent 送信 引数 ・ 第一引数: intent ・ 第二引数: 任意のリクエストコード(int 型)
------------------------	---

Android 版 JPKI 利用者ソフトからの情報を onActivityResult(int requestCode, int resultCode, Intent data)で受け取る

data.getIntExtra	コマンドタイプを取得 引数 ・ 第一引数: “command_type”
------------------	--

data.getByteArrayExtra	署名値の取得 引数 ・ 第一引数: “signature”
------------------------	-------------------------------------

data.getIntExtra	エラーコードを取得(int 型) 引数 ・ 第一引数: “err_code” 取得した値が 0000 の場合は正常に処理された
------------------	--

data.getIntExtra	詳細コードを取得(int 型) 引数 ・ 第一引数: “detail_code” 取得した値が 00000000 の場合は正常に処理された
------------------	--

終了処理	2 終了処理 終了処理 参照
------	--

(4) 署名生成処理(ダイジェスト[ハッシュ値]を入力する場合)

初期処理	1 初期処理 初期処理 参照
------	--

Intent インスタンスの生成	引数 ・ 第一引数: Intent.ACTION_SEND
------------------	----------------------------------

intent.setClassName	JPKI 利用者ソフトのパッケージ名とクラス名の設定
---------------------	----------------------------

	引数 <ul style="list-style-type: none"> ・ 第一引数: "jp.go.jpki.mobile.utility" ・ 第二引数: "jp.go.jpki.mobile.intent.JPKIIntentActivity"
intent.putExtra	拡張データの設定(コマンドタイプの設定) 引数 <ul style="list-style-type: none"> ・ 第一引数: "command_type" ・ 第二引数: 0x01001004
intent.putExtra	拡張データの設定(ダイジェスト(ハッシュ値)の設定) 引数 <ul style="list-style-type: none"> ・ 第一引数: "hash" ・ 第二引数: byte[]型
intent.putExtra	拡張データの設定(ハッシュアルゴリズムの指定) 引数 <ul style="list-style-type: none"> ・ 第一引数: "alg_id" ・ 第二引数: int 型
startActivityForResult	Android 版 JPKI 利用者ソフトの起動と intent 送信 引数 <ul style="list-style-type: none"> ・ 第一引数: intent ・ 第二引数: 任意のリクエストコード(int 型)
Android 版 JPKI 利用者ソフトからの情報を onActivityResult(int requestCode, int resultCode, Intent data)で受け取る	
data.getIntExtra	コマンドタイプを取得 引数 <ul style="list-style-type: none"> ・ 第一引数: "command_type"
data.getByteArrayExtra	署名値の取得 引数 <ul style="list-style-type: none"> ・ 第一引数: "signature"
data.getIntExtra	エラーコードを取得(int 型) 引数 <ul style="list-style-type: none"> ・ 第一引数: "err_code" 取得した値が 0000 の場合は正常に処理された

data.getIntExtra	詳細コードを取得(int 型) 引数 ・ 第一引数: “ detail_code ” 取得した値が 00000000 の場合は正常に処理された
------------------	--

終了処理	2 終了処理終了処理参照
------	------------------------------

(5) 署名検証処理(検証対象データを渡す場合)

「第 5 章 第 3 節 3 (5) [署名検証処理\(検証対象データを渡す場合\)](#)署名検証処理(検証対象データを渡す場合)」を参照。

(6) 署名検証処理(ダイジェスト[ハッシュ値]を入力する場合)

「第 5 章 第 3 節 3 (6) 署名検証処理(ダイジェスト[ハッシュ値]を入力する場合)」を参照。

(7) 繰り返し署名生成処理(署名対象データを渡すパターン)

繰り返し生成処理に関しては、「(3) [署名生成処理\(署名対象データを渡す場合\)](#)署名生成処理(署名対象データを渡す場合)」網掛け部分を署名対象データの個数分だけ繰り返して呼び出す。

(8) 繰り返し署名生成処理(ダイジェスト[ハッシュ値]を渡すパターン)

繰り返し生成処理に関しては、「(4) [署名生成処理\(ダイジェスト\[ハッシュ値\]を入力する場合\)](#)署名生成処理(ダイジェスト[ハッシュ値]を入力する場合)」網掛け部分を署名対象データの個数分だけ繰り返して呼び出す。

5 個人認証サービス AP(署名用、利用者証明用共通)

(1) 証明書表示

初期処理	1 初期処理 初期処理 参照
Intent インスタンスの生成	引数 ・ 第一引数: Intent.ACTION_SEND
intent.setClassName	JPKI 利用者ソフトのパッケージ名とクラス名の設定 引数 ・ 第一引数: “jp.go.jpki.mobile.utility” ・ 第二引数: “jp.go.jpki.mobile.intent.JPKIIntentActivity”
intent.putExtra	拡張データの設定(コマンドタイプの設定) 引数 ・ 第一引数: “command_type” ・ 第二引数: 0x01003001
intent.putExtra	拡張データの設定(証明書データ(DER形式)) 引数 ・ 第一引数: “cert” ・ 第二引数: byte[]型
startActivityForResult	Android 版 JPKI 利用者ソフトの起動と intent 送信 引数 ・ 第一引数: intent ・ 第二引数: 任意のリクエストコード(int型)
Android 版 JPKI 利用者ソフトからの情報を onActivityResult(int requestCode, int resultCode, Intent data)で受け取る	
data.getIntExtra	コマンドタイプを取得 引数 ・ 第一引数: “command_type”
data.getIntExtra	エラーコードを取得(int型) 引数 ・ 第一引数: “err_code”

	取得した値が 0000 の場合は正常に処理された
--	--------------------------

data.getIntExtra	詳細コードを取得(int 型) 引数 ・ 第一引数: “ detail_code ” 取得した値が 00000000 の場合は正常に処理された
------------------	--

終了処理	2 終了処理終了処理参照
------	------------------------------

(2) 基本 4 情報取得

初期処理	1 初期処理初期処理参照
------	------------------------------

Intent インスタンスの生成	引数 ・ 第一引数: Intent.ACTION_SEND
------------------	----------------------------------

intent.setClassName	JPKI 利用者ソフトのパッケージ名とクラス名の設定 引数 ・ 第一引数: “ jp.go.jpki.mobile.utility ” ・ 第二引数: “ jp.go.jpki.mobile.intent.JPKIIntentActivity ”
---------------------	---

intent.putExtra	拡張データの設定(コマンドタイプの設定) 引数 ・ 第一引数: “ command_type ” ・ 第二引数: 0x01003002
-----------------	--

intent.putExtra	拡張データの設定(証明書取得で取得したデータ(DER 形式)) 引数 ・ 第一引数: “ cert ” ・ 第二引数: byte[]型
-----------------	--

startActivityForResult	Android 版 JPKI 利用者ソフトの起動と intent 送信 引数 ・ 第一引数: intent ・ 第二引数: 任意のリクエストコード(int 型)
------------------------	---

Android 版 JPKI 利用者ソフトからの情報を onActivityResult(int requestCode, int resultCode, Intent data)で受け取る	
---	--

data.getIntExtra	コマンドタイプを取得 引数 ・ 第一引数: “ command_type ”
data.getStringExtra	住所を取得 引数 ・ 第一引数: “ address ”
data.getStringExtra	生年月日を取得 引数 ・ 第一引数: “ date_of_birth ”
data.getStringExtra	性別を取得 引数 ・ 第一引数: “ gender ”
data.getStringExtra	氏名を取得 引数 ・ 第一引数: “ name ”
data.getStringExtra	住所の代替文字を取得 引数 ・ 第一引数: “ substitute_character_of_address ”
data.getStringExtra	氏名の代替文字を取得 引数 ・ 第一引数: “ substitute_character_of_name ”
data.getIntExtra	エラーコードを取得(int 型) 引数 ・ 第一引数: “ err_code ” 取得した値が 0000 の場合は正常に処理された
data.getIntExtra	詳細コードを取得(int 型) 引数 ・ 第一引数: “ detail_code ” 取得した値が 00000000 の場合は正常に処理された
終了処理	2 終了処理終了処理参照

(3) 官職証明書検証

初期処理	1 初期処理 初期処理 参照
Intent インスタンスの生成	引数 ・ 第一引数: Intent.ACTION_SEND
intent.setClassName	JPKI 利用者ソフトのパッケージ名とクラス名の設定 引数 ・ 第一引数: “jp.go.jpki.mobile.utility” ・ 第二引数: “jp.go.jpki.mobile.intent.JPKIIntentActivity”
intent.putExtra	拡張データの設定(コマンドタイプの設定) 引数 ・ 第一引数: “command_type” ・ 第二引数: 0x01003003
intent.putExtra	拡張データの設定(証明書データ(DER 形式)) 引数 ・ 第一引数: “cert” ・ 第二引数: byte[]型
startActivityForResult	Android 版 JPKI 利用者ソフトの起動と intent 送信 引数 ・ 第一引数: intent ・ 第二引数: 任意のリクエストコード(int 型)
Android 版 JPKI 利用者ソフトからの情報を onActivityResult(int requestCode, int resultCode, Intent data)で受け取る	
data.getIntExtra	コマンドタイプを取得 引数 ・ 第一引数: “command_type”
data.getIntExtra	官職証明書検証機能で取得した証明書の検証の結果を取得 引数 ・ 第一引数: “cert_status”

data.getIntExtra	官職証明書検証機能で取得した認証パス検証結果コードを取得 引数 ・ 第一引数: “ cert_path_status ”
------------------	--

data.getIntExtra	官職証明書検証機能で取得した有効性検証結果コードを取得 引数 ・ 第一引数: “ response_status ”
------------------	--

data.getIntExtra	エラーコードを取得(int 型) 引数 ・ 第一引数: “ err_code ” 取得した値が 0000 の場合は正常に処理された
------------------	--

data.getIntExtra	詳細コードを取得(int 型) 引数 ・ 第一引数: “ detail_code ” 取得した値が 00000000 の場合は正常に処理された
------------------	--

終了処理	2 終了処理終了処理参照
------	------------------------------

(4) 自分の電子証明書の有効性確認

初期処理	1 初期処理初期処理参照
------	------------------------------

Intent インスタンスの生成	引数 ・ 第一引数: Intent.ACTION_SEND
------------------	----------------------------------

intent.setClassName	JPKI 利用者ソフトのパッケージ名とクラス名の設定 引数 ・ 第一引数: “ jp.go.jpki.mobile.utility ” ・ 第二引数: “ jp.go.jpki.mobile.intent.JPKIIntentActivity ”
---------------------	---

intent.putExtra	拡張データの設定(コマンドタイプの設定) 引数 ・ 第一引数: “ command_type ” ・ 第二引数: 0x01003004
-----------------	--

intent.putExtra	拡張データの設定(証明書取得で取得したデータ(DER 形式)) 引数
-----------------	---------------------------------------

	<ul style="list-style-type: none"> ・ 第一引数: “ cert ” ・ 第二引数: byte[]型
startActivityForResult	<p>Android 版 JPKI 利用者ソフトの起動と intent 送信 引数</p> <ul style="list-style-type: none"> ・ 第一引数: intent ・ 第二引数: 任意のリクエストコード(int 型)
<p>Android 版 JPKI 利用者ソフトからの情報を onActivityResult(int requestCode, int resultCode, Intent data)で受け取る</p>	
data.getIntExtra	<p>コマンドタイプを取得 引数</p> <ul style="list-style-type: none"> ・ 第一引数: “ command_type ”
data.getIntExtra	<p>有効性確認結果を取得 引数</p> <ul style="list-style-type: none"> ・ 第一引数: “ code ”
data.getStringExtra	<p>有効性確認中に発生した Exception のメッセージを取得 引数</p> <ul style="list-style-type: none"> ・ 第一引数: “ exception ”
data.getStringExtra	<p>確認結果メッセージを取得 引数</p> <ul style="list-style-type: none"> ・ 第一引数: “ confirm_msg ”
data.getIntExtra	<p>エラーコードを取得(int 型) 引数</p> <ul style="list-style-type: none"> ・ 第一引数: “ err_code ” <p>取得した値が 0000 の場合は正常に処理された</p>
data.getIntExtra	<p>詳細コードを取得(int 型) 引数</p> <ul style="list-style-type: none"> ・ 第一引数: “ detail_code ” <p>取得した値が 00000000 の場合は正常に処理された</p>
終了処理	<p>2 終了処理終了処理参照</p>

(5) ICカード種別取得

初期処理	1 初期処理初期処理参照
Intent インスタンスの生成	引数 ・ 第一引数: Intent.ACTION_SEND
intent.setClassName	JPKI 利用者ソフトのパッケージ名とクラス名の設定 引数 ・ 第一引数: “jp.go.jpki.mobile.utility” ・ 第二引数: “jp.go.jpki.mobile.intent.JPKIIntentActivity”
intent.putExtra	拡張データの設定(コマンドタイプの設定) 引数 ・ 第一引数: “command_type” ・ 第二引数: 0x01003005
startActivityForResult	Android 版 JPKI 利用者ソフトの起動と intent 送信 引数 ・ 第一引数: intent ・ 第二引数: 任意のリクエストコード(int 型)
Android 版 JPKI 利用者ソフトからの情報を onActivityResult(int requestCode, int resultCode, Intent data)で受け取る	
data.getIntExtra	コマンドタイプを取得 引数 ・ 第一引数: “command_type”
data.getIntExtra	IC カード種別を取得 引数 ・ 第一引数: “id”
data.getStringExtra	IC カード token 情報を取得 引数 ・ 第一引数: “token”
data.getIntExtra	エラーコードを取得(int 型) 引数

	<ul style="list-style-type: none">・ 第一引数: “err_code” 取得した値が 0000 の場合は正常に処理された
data.getIntExtra	詳細コードを取得(int 型) 引数 <ul style="list-style-type: none">・ 第一引数: “detail_code” 取得した値が 00000000 の場合は正常に処理された
終了処理	2 終了処理終了処理参照

第6章 画面仕様
第1節 画面一覧

表 6-1 画面一覧

#	機能名	画面名		概要
1	証明書表示機能	基本画面	署名用電子証明書	署名用電子証明書の基本情報を表示する。 [有効性確認]ボタンの押下により、自分の電子証明書の有効性確認を行う。 [ファイル出力]ボタンの押下により、電子証明書の内容をファイルに出力する。
2			利用者証明用電子証明書	利用者証明用電子証明書の基本情報を表示する。 [有効性確認]ボタンの押下により、自分の電子証明書の有効性確認を行う。 [ファイル出力]ボタンの押下により、電子証明書の内容をファイルに出力する。
3			認証局の自己署名証明書	認証局の自己署名証明書(ルート認証局の自己署名証明書, リンク証明書, 下位認証局の自己署名証明書, 相互認証証明書, 自己署名証明書等)の基本情報を表示する。 [ファイル出力]ボタンの押下により、電子証明書の内容をファイルに出力する。
4			その他の電子証明書	上記以外の証明書(官職証明書, 職責証明書, その他の証明書)の基本情報を表示する。 [証明書検証]ボタン押下により、証明書の検証を行う。 [ファイル出力]ボタンの押下により、電子証明書の内容をファイルに出力する。
5		詳細画面	電子証明書内の全ての記載事項を表示する。 署名用電子証明書または利用者証明用電子証明書の場合、[有効性確認]ボタンの押下により、自分の電子証明書の有効性確認を行う。 その他の電子証明書の場合、[証明書検証]ボタン押下により、証明書の検証を行う。 [ファイル出力]ボタンの押下により、電子証明書の内容をファイルに出力する。	
6		官職証明書・職責証明書検証画面	その他の電子証明書(基本/詳細)画面にて、[証明書検証]ボタンを押下し、証明書検証を行い、証明書検証結果を表示する。	

7		利用者証明書有効性確認画面	署名用電子証明書(基本/詳細)画面または利用者証明用電子証明書(基本/詳細)画面にて、[有効性確認]ボタンを押下して有効性確認を行い、有効性確認結果を表示する。
8	その他	パスワード入力画面	パスワード入力の案内を行う。
9		ICカードセット案内画面	ICカードセットの案内を行う。

第 2 節 署名用電子証明書基本画面

画面名	署名用電子証明書基本画面	
概要	署名用電子証明書の基本情報を表示する。	
画面レイアウト		
		
<p>表示項目と証明書領域の対応は、表 6 - 2 表示項目と証明書領域の対応（署名用電子証明書） 表 6 - 2 表示項目と証明書領域の対応（署名用電子証明書） を参照。</p>		
画面項目説明		
#	項目名	概要
	戻るボタン	署名用電子証明書基本画面を閉じ、前画面に戻る。
	タイトルバー	表示テキストは「電子証明書表示」とする。
	タイトル	表示する電子証明書の種類を表示する。 表示テキストは以下の通りとする。 「公的個人認証サービス 利用者の署名用電子証明書」
	タブ	基本画面と詳細画面を切り替える。

氏名 1-4	利用者の氏名を表示する。大括弧「[]」内は旧氏を示す。括弧「()」内は通称を示す。1行で収まらない場合は、折り返して表示する。(単語途中であっても折り返しを行う。)
代替文字の使用 (氏名) 1-4	氏名の代替文字を表示する。代替文字がない場合は、表示しない。
生年月日	利用者の生年月日を和暦で表示する。
性別	利用者の性別(男/女/不明)を表示する。
住所 1-4	利用者の住所を表示する。 1行で収まらない場合は、折り返して表示する。(単語途中であっても折り返しを行う。)
代替文字の使用 (住所) 1-4	住所の代替文字を表示する。代替文字がない場合は、表示しない。
発行年月日	電子証明書の発行年月日を和暦で表示する。
有効期間の満了日	電子証明書の有効期間の満了日を和暦で表示する。
発行者	電子証明書の発行者を表示する。
ファイル出力ボタン	[ファイル出力]ボタンを押下することによって電子証明書の内容をファイルに出力する。出力様式は以下の2通り。 ・電子証明書を DER 形式のファイルとして出力する。拡張子は *.cer とする。 ・画面表示内容をテキスト形式のファイルとして出力する。拡張子は *.txt とする。
ステータスのタイトル	「有効性確認結果」を表示する。
ステータス	有効性確認状態を以下のように表示する。 確認前：電子証明書の有効性確認は行われていません。 確認後：有効性確認結果：「有効」 有効性確認結果：「有効期間切れ」 有効性確認結果：「失効済み」 有効性確認結果：「失効申請受理済み」 有効性確認結果：「一時保留」 有効性確認結果：「確認失敗(xxxxxx)」 「xxxxxx」は原因を示すエラーコード。詳細は、 表 6-10 表 6-10 有効性確認結果「確認失敗」のエラーコード一覧 を参照。

有効性確認ボタン	[有効性確認]ボタンを押下することによって、公的個人認証サービス AP の自分の電子証明書の有効性確認機能呼び出し、オンライン窓口サービスに有効性確認の問い合わせを行う。自分の電子証明書の有効性確認機能の戻り値を判断して有効性確認結果画面を表示する。
閉じるボタン	署名用電子証明書基本画面を閉じ、前画面に戻る。
ヘルプボタン	オンラインヘルプ表示機能を実行する。
バックキー	署名用電子証明書基本画面を閉じ、前画面に戻る。

1については表 6-2表-6-2および第6章 第11節 設定ルール一覧を参照。

表 6-2 表示項目と証明書領域の対応（署名用電子証明書）

#	項目名	証明書の項目名		表示方法
		上位項目名	項目名	
1	氏名	SubjectAltName	commonName	設定値をそのまま表示。 大括弧「 [] 」内は旧氏を示す。 括弧「 () 」内は通称を示す。
2	代替文字の使用(氏名)		substituteCharacterOf-CommonName ¹	・代替文字を「鍵括弧」付で表示。 ・代替文字が複数ある場合は代替文字を続けて表示。 例)「吉」「郎」は代替文字です。
3	生年月日		dateOfBirth ²	設定値を和暦に変換して表示。
4	性別		gender ³	設定値を日本語表記に変換して表示。
5	住所		address	設定値をそのまま表示。
6	代替文字の使用(住所)		substituteCharacterOf-Address ¹	・代替文字を「鍵括弧」付で表示。 ・代替文字が複数ある場合は代替文字を続けて表示。 例)「葛」「飾」は代替文字です。
7	発行年月日	Validity	notBefore	設定値を和暦(日本標準時)に変換して表示。書式は「GGYY年MM月DD日」(GGは元号、時分秒は表示せず。)
8	有効期間の満了日		notAfter	
9	発行者	IssuerAltName	organizationalUnitName	設定値をそのまま表示。

1～3については第6章 第11節 設定ルール一覧を参照。

第 3 節 利用者証明用電子証明書基本画面

画面名	利用者証明用電子証明書基本画面	
概要	利用者証明用電子証明書の基本情報を表示する。	
画面レイアウト		
		
表示項目と証明書領域の対応は、 表 6-3 表-6-3 表示項目と証明書領域の対応(利用者証明用電子証明書)を参照。		
画面項目説明		
#	項目名	概要
	戻るボタン	利用者証明用電子証明書基本画面を閉じ、前画面に戻る。
	タイトルバー	表示テキストは「電子証明書表示」とする。
	タイトル	表示する電子証明書の種類を表示する。 表示テキストは以下の通りとする。 「公的個人認証サービス 利用者の利用者証明用電子証明書」
	タブ	基本画面と詳細画面を切り替える。
	主体者	主体者を表示する。 設定値が1行で収まらない場合は、折り返して表示する。 (単語途中であっても折り返しを行う。)
	発行年月日	電子証明書の発行年月日を和暦で表示する。

有効期間の満了日	電子証明書の有効期間の満了日を和暦で表示する。
発行者	電子証明書の発行者を表示する。 設定値が 1 行で収まらない場合は、折り返して表示する。 (単語途中であっても折り返しを行う。)
ファイル出力ボタン	[ファイル出力]ボタンを押下することによって電子証明書の内容をファイルに出力する。出力様式は以下の 2 通り。 ・電子証明書を DER 形式のファイルとして出力する。拡張子は*.cer とする。 ・画面表示内容をテキスト形式のファイルとして出力する。拡張子は*.txt とする。
ステータスのタイトル	「有効性確認結果」を表示する。
ステータス	有効性確認状態を以下のように表示する。 確認前：電子証明書の有効性確認は行われていません。 確認後：有効性確認結果：「有効」 有効性確認結果：「有効期間切れ」 有効性確認結果：「失効済み」 有効性確認結果：「失効申請受理済み」 有効性確認結果：「一時保留」 有効性確認結果：「確認失敗(xxxxxx)」 「xxxxxx」は原因を示すエラーコード。詳細は、 表 6-10 表 6-10 有効性確認結果「確認失敗」のエラーコード一覧を参照。
有効性確認ボタン	[有効性確認]ボタンを押下することによって、公的個人認証サービス AP の自分の電子証明書の有効性確認機能呼び出し、オンライン窓口サービスに有効性確認の問い合わせを行う。 自分の電子証明書の有効性確認機能の戻り値を判断して有効性確認結果画面を表示する。
閉じるボタン	利用者証明用電子証明書基本画面を閉じ、前画面に戻る。
ヘルプボタン	オンラインヘルプ表示機能を実行する。
バックキー	利用者証明用電子証明書基本画面を閉じ、前画面に戻る。

表 6 - 3 表示項目と証明書領域の対応 (利用者証明用電子証明書)

#	項目名	証明書の項目名		表示方法
		上位項目名	項目名	
1	主体者	SubjectAltName または Subject	CountryName CommonName 例)利用者証明用電子証明書の場合	SubjectAltName の DN を全て表示。 DN は属性毎に改行して表示し、「(属性名の略語) = (設定値)」にて表記。 subjectAltName に記述がない場合は subject を表示。
2	発行年月日	Validity	notBefore	設定値を和暦(日本標準時)に変換して表示。書式は「GGYY年MM月DD日」(GGは元号、時分秒は表示せず。)
3	有効期間の満了日		notAfter	
4	発行者	IssuerAltName または Issuer	CountryName OrganizationName OrganizationalUnitName OrganizationalUnitName 例)利用者証明用電子証明書の場合	IssuerAltName の DN を全て表示。 DN は属性毎に改行して表示し、「(属性名の略語) = (設定値)」にて表記。issuerAltName に記述がない場合は issuer を表示。

第 4 節 認証局の自己署名証明書基本画面

画面名	認証局の自己署名証明書基本画面	
概要	認証局の自己署名証明書の基本情報を表示する。	
画面レイアウト		
		
表示項目と証明書領域の対応は、表 6-4 表 6-4 表示項目と証明書領域の対応（認証局の自己署名証明書）を参照。		
画面項目説明		
#	項目名	概要
	戻るボタン	認証局の自己署名証明書基本画面を閉じ、前画面に戻る。
	タイトルバー	表示テキストは「電子証明書表示」とする。
	タイトル	表示する電子証明書の種類を表示する。 表示テキストは以下の通りとする。 ・署名用認証局の自己署名証明書の場合 「公的個人認証サービス 署名用認証局の電子証明書」 ・利用者証明用認証局の自己署名証明書の場合 「公的個人認証サービス 利用者証明用認証局の電子証明書」
	タブ	基本画面と詳細画面を切り替える。

主体者	主体者を表示する。 設定値が 1 行で収まらない場合は、折り返して表示する。(単語途中であっても折り返しを行う。)
発行年月日	電子証明書の発行年月日を和暦で表示する。
有効期間の満了日	電子証明書の有効期間の満了日を和暦で表示する。
発行者	電子証明書の発行者を表示する。 設定値が 1 行で収まらない場合は、折り返して表示する。(単語途中であっても折り返しを行う。)
フィンガープリント	電子証明書のハッシュ値を計算して表示。ハッシュ関数は「sha256」を使用する。
ファイル出力ボタン	[ファイル出力]ボタンを押下することによって電子証明書の内容をファイルに出力する。出力様式は以下の 2 通り。 ・電子証明書を DER 形式のファイルとして出力する。拡張子は*.cer とする。 ・画面表示内容をテキスト形式のファイルとして出力する。拡張子は*.txt とする。
閉じるボタン	認証局の自己署名証明書基本画面を閉じ、前画面に戻る。
ヘルプボタン	オンラインヘルプ表示機能を実行する。
バックキー	認証局の自己署名証明書基本画面を閉じ、前画面に戻る。

表 6 - 4 表示項目と証明書領域の対応 (認証局の自己署名証明書)

#	項目名	証明書の項目名		表示方法
		上位項目名	項目名	
1	主体者	SubjectAltName または Subject	CountryName OrganizationName OrganizationalUnitName OrganizationalUnitName 例) 認証局の自己署名証明書の場合	SubjectAltName の DN を全て表示。 DN は属性毎に改行して表示し、「(属性名の略語) = (設定値)」にて表記。 subjectAltName に記述がない場合は subject を表示。
2	発行年月日	Validity	notBefore	設定値を和暦(日本標準時)に変換して表示。書式は「GGYY年MM月DD日」(GGは元号、時分秒は表示せず。)
3	有効期間の満了日		notAfter	
4	発行者	IssuerAltName または Issuer	CountryName OrganizationName OrganizationalUnitName OrganizationalUnitName 例) 認証局の自己署名証明書の場合	IssuerAltName の DN を全て表示。 DN は属性毎に改行して表示し、「(属性名の略語) = (設定値)」にて表記。 issuerAltName に記述がない場合は issuer を表示。

5	フィンガ ープリン ト	-	-	電子証明書のハッシュ値を計 算して表示。ハッシュ関数は 「sha256」を使用する。
---	-------------------	---	---	--

第5節 その他の電子証明書基本画面

画面名	その他の電子証明書基本画面
概要	利用者証明書、認証局の自己署名証明書以外の電子証明書の基本情報を表示する。

画面レイアウト



表示項目と証明書領域の対応は、[表 6-5](#)表 6-5 表示項目と証明書領域の対応（官職証明書 / 職責証明書 / その他の証明書）を参照。

画面項目説明

#	項目名	概要
	戻るボタン	その他の電子証明書基本画面を閉じ、前画面に戻る。
	タイトルバー	表示テキストは「電子証明書表示」とする。
	タイトル	表示する電子証明書の種類を表示する。 表示テキストは以下の通りとする。 <ul style="list-style-type: none"> ・ 認証局の自己署名証明書の場合 「認証局の電子証明書」 ・ 上記以外の電子証明書の場合 「電子証明書」
	タブ	基本画面と詳細画面を切り替える。

主体者	主体者を表示する。 設定値が 1 行で収まらない場合は、折り返して表示する。(単語途中であっても折り返しを行う。)
発行年月日	電子証明書の発行年月日を和暦で表示する。
有効期間の満了日	電子証明書の有効期間の満了日を和暦で表示する。
発行者	電子証明書の発行者を表示する。 設定値が 1 行で収まらない場合は、折り返して表示する。(単語途中であっても折り返しを行う。)
ファイル出力ボタン	[ファイル出力]ボタンを押下することによって電子証明書の内容をファイルに出力する。出力様式は以下の 2 通り。 ・電子証明書を DER 形式のファイルとして出力する。拡張子は *.cer とする。 ・画面表示内容をテキスト形式のファイルとして出力する。拡張子は *.txt とする。
ステータスのタイトル	「証明書検証結果」を表示する。
ステータス	証明書検証状態を以下のように表示する。 検証前：電子証明書の検証は行われていません。 検証後：証明書検証結果「有効」 証明書検証結果「無効 (xxxxxx)」 証明書検証結果「検証失敗 (xxxxxx)」 「xxxxxx」は原因を示すエラーコード。詳細は、 表 6-8 表 6-8 証明書検証結果「無効」の場合のエラーコード一覧 、 表 6-9 表 6-9 証明書検証結果「検証失敗」の場合のエラーコード一覧 を参照。
証明書検証ボタン	[証明書検証]ボタンを押下することによって、公的個人認証サービス AP の官職証明書検証機能呼び出し、官職証明書検証サービスに証明書検証の問い合わせを行う。 官職証明書検証機能の戻り値を判断して官職証明書・職責証明書検証画面を表示する。
閉じるボタン	その他の電子証明書基本画面を閉じ、前画面に戻る。
ヘルプボタン	オンラインヘルプ表示機能を実行する。
バックキー	その他の電子証明書基本画面を閉じ、前画面に戻る。

表 6-5 表示項目と証明書領域の対応（官職証明書 / 職責証明書 / その他の証明書）

#	項目名	証明書の項目名		表示方法
		上位項目名	項目名	
1	主体者	SubjectAltName または Subject	CountryName OrganizationName OrganizationalUnitName CommonName 例)官職証明書の場合	SubjectAltName の DN を全て表示。 DN は属性毎に改行して表示し、「(属性名の略語) = (設定値)」にて表記。 subjectAltName に記述がない場合は subject を表示。
2	発行年月日	Validity	notBefore	設定値を和暦(日本標準時)に変換して表示。書式は「GGYY年MM月DD日」(GGは元号、時分秒は表示せず。)
3	有効期間の満了日		notAfter	
4	発行者	IssuerAltName または Issuer	CountryName OrganizationName OrganizationalUnitName 例)官職証明書の場合	IssuerAltName の DN を全て表示。 DN は属性毎に改行して表示し、「(属性名の略語) = (設定値)」にて表記。 issuerAltName に記述がない場合は issuer を表示。
5	フィンガープリント	-	-	認証局の自己署名証明書の場合のみ表示。 電子証明書のハッシュ値を計算して表示。ハッシュ関数は「sha1」または「sha256」を使用する。

第6節 詳細画面

画面名	詳細画面	
概要	電子証明書内の全ての記載事項を表示する。	
画面レイアウト		
		
表示項目と証明書領域の対応は、 表 6-6 表-6-6項目名と証明書基本領域との対応、 表 6-7 表-6-7項目名と証明書標準拡張領域との対応を参照。		
画面項目説明		
#	項目名	概要
	戻るボタン	詳細画面を閉じ、前画面に戻る。
	タイトルバー	表示テキストは「電子証明書表示」とする。

タイトル	<p>表示する電子証明書の種類を表示する。</p> <p>表示テキストは以下の通りとする。</p> <ul style="list-style-type: none"> ・ 住基カードに格納された署名用電子証明書および個人番号カードに格納された署名用電子証明書の場合 「公的個人認証サービス 利用者の署名用電子証明書」 ・ 個人番号カードに格納された利用者証明用電子証明書の場合 「公的個人認証サービス 利用者の利用者証明用電子証明書」 ・ 都道府県知事の自己署名証明書および署名用認証局の自己署名証明書の場合 「公的個人認証サービス 署名用認証局の電子証明書」 ・ 利用者証明用認証局の自己署名証明書の場合 「公的個人認証サービス 利用者証明用認証局の電子証明書」 ・ 上記以外の認証局の自己署名証明書の場合 「認証局の電子証明書」 ・ 上記以外の証明書の場合 「電子証明書」
タブ	基本画面と詳細画面を切り替える。
フィンガープリント	<p>電子証明書のハッシュ値を計算して表示する。</p> <p>ハッシュ関数は「sha1」または「sha256」を使用する。</p>
証明書内容表示領域	<p>電子証明書内の記載事項を表示する。</p> <p>フィールド名と値の詳細は、表 6-6 項目名と証明書基本領域との対応表—6-6 項目名と証明書基本領域との対応と表 6-7 項目名と証明書標準拡張領域との対応表—6-7 項目名と証明書標準拡張領域との対応の基本画面を参照。</p>
ファイル出力ボタン	<p>[ファイル出力]ボタンを押下することによって電子証明書の内容をファイルに出力する。出力様式は以下の2通り。</p> <ul style="list-style-type: none"> ・ 電子証明書を DER 形式のファイルとして出力する。拡張子は*.cer とする。 ・ 画面表示内容をテキスト形式のファイルとして出力する。拡張子は*.txt とする。
ステータスのタイトル	<p>表示テキストは以下の通りとする。</p> <ul style="list-style-type: none"> ・ 利用者証明書の場合 「有効性確認結果」 ・ 官職証明書 / 職責証明書の場合 「証明書検証結果」

ステータス	利用者証明書の場合は有効性確認結果を表示する。(ステータスの概要については、第2節 署名用電子証明書基本画面署名用電子証明書基本画面 の画面項目説明を参照。) 官職証明書 / 職責証明書の場合は証明書確認結果を表示する。(ステータスの概要については、第5節 その他の電子証明書基本画面 の画面項目説明を参照。)
有効性確認ボタン / 証明書検証ボタン	利用者証明書の場合は有効性確認ボタンを表示する。(有効性確認機能については、第2節 署名用電子証明書基本画面署名用電子証明書基本画面 の画面項目説明を参照。) 官職証明書 / 職責証明書の場合は証明書検証ボタンを表示する。(証明書検証機能については、第5節 その他の電子証明書基本画面 の画面項目説明を参照。)
閉じるボタン	詳細画面を閉じ、前画面に戻る。
ヘルプボタン	オンラインヘルプ表示機能を実行する。
バックキー	その他の電子証明書基本画面を閉じ、前画面に戻る。

表 6 - 6 項目名と証明書基本領域との対応

#	項目名	証明書の項目名		表示方法
		上位項目名	項目名	
1	バージョン	version		Version 表記。Version3 は “ V3 ” とする。
2	シリアル番号	serialNumber		設定値をそのまま表示。
3	署名アルゴリズム	signature	algorithm parameters	algorithm の OID を RFC の規定値に変換した値。
4	発行者	issuer	countryName organizationName organizationalUnitName 等	DN を属性毎に改行。各行は「(属性の略語) = (設定値)」にて表記。 <属性の略語> 「countryName」 C 「organizationName」 O 「organizationalUnitName」 OU
5	発行年月日	Validity	notBefore	設定値を西暦(日本標準時)で表示。書式は「YYYY年MM月DD日hh時mm分ss秒」。
6	有効期間の満了日		notAfter	

#	項目名	証明書の項目名		表示方法
		上位項目名	項目名	
7	主体者 4	subject	countryName localityName commonName 等	DNを属性毎に改行。各行は「(属性の略語)=(設定値)」にて表記。 <属性の略語> 「countryName」 C 「localityName」 L 「commonName」 CN commonNameが 発行要求発生時刻の場合は、設定値をそのまま表示。
8	発行申請送信時刻 5	subject	commonName のみが表示対象	commonNameから発行申請送信時刻を抜き出して表示。
9	受付端末識別記号 5			commonNameから受付端末識別記号を抜き出して表示。
10	発行申請送信時刻 6	subject	commonName のみが表示対象	commonNameから発行申請送信時刻を抜き出して表示。
11	シーケンス番号 6			commonNameからシーケンス番号を抜き出して表示。
12	受付端末識別記号 6			commonNameから受付端末識別記号を抜き出して表示。
13	ランダム文字列 7	subject	commonName のみが表示対象	commonNameからランダム文字列を抜き出して表示。
14	受付端末識別記号 7			commonNameから受付端末識別記号を抜き出して表示。
15	主体者の公開鍵情報	subjectPublicKeyInfo	algorithm subjectPublicKey	SubjectPublicKeyの設定値(16進数)をそのまま表示。
16	発行者ユニーク識別子	issuerUniqueID		設定値をそのまま表示。
17	主体者ユニーク識別子	subjectUniqueID		設定値をそのまま表示。

4：利用者証明書以外の場合

5：住基カードに格納された署名用電子証明書の場合

6：個人番号カードに格納された署名用電子証明書の場合

7：個人番号カードに格納された利用者証明用電子証明書の場合

表 6-7 項目名と証明書標準拡張領域との対応

#	項目名	証明書の項目名		表示方法
		上位項目名	項目名	
1	認証局鍵識別子	AuthorityKeyIdentifier	keyIdentifier authorityCertIssuer authorityCertSerialNumber	AuthorityKeyIdentifier の情報を項目毎に階層表示。なお、表示の際は以下のルールを適用する。 ・項目名は英語のまま。 ・OID は RFC の規定値に変換。 ・DN は属性毎に改行。
2	鍵用途	KeyUsage	digitalSignature nonRepudiation keyEncipherment dataEncipherment keyAgreement keyCertSign cRLSign encipherOnly decipherOnly	KeyUsage の情報を項目毎に階層表示。なお、表示の際は以下のルールを適用する。 ・項目名は英語のまま。 ・OID は RFC の規定値に変換。 ・KeyUsage のビット列は、鍵用途の英語名に変換してカンマ区切り表示。
3	主体者代替名	SubjectAltName	countryName organizationName organizationalUnitName 等	SubjectAltName の情報を項目毎に階層表示。なお、表示の際は以下のルールを適用する。 ・項目名は英語のまま。 ・OID は RFC の規定値に変換。
4	発行者代替名	issuerAltName	countryName organizationName organizationalUnitName 等	issuerAltName の情報を項目毎に階層表示。なお、表示の際は以下のルールを適用する。 ・項目名は英語のまま。 ・OID は RFC の規定値に変換。
5	基本制約	BasicConstraints	capathLenConstraint	BasicConstraints の情報を項目毎に階層表示。なお、表示の際は以下のルールを適用する。 ・項目名は英語のまま。 ・OID は RFC の規定値に変換。
6	CRL 分配点	CRLDistributionPoints	countryName organizationName organizationalUnitName 等	CRLDistributionPoints の情報を項目毎に階層表示。なお、表示の際は以下のルールを適用する。 ・項目名は英語のまま。 ・OID は RFC の規定値に変換。 ・DN は属性毎に改行。

#	項目名	証明書項目名		表示方法
		上位項目名	項目名	
7	証明書ポリシー	Certificate Policies	policyIdentifier policyQualifiers	CertificatePolicies の情報を項目毎に階層表示。なお、表示の際は以下のルールを適用する。 ・項目名は英語のまま。 ・OID は RFC の規定値に変換。
8	主体者鍵識別子	SubjectKeyIdentifier	keyIdentifier	SubjectKeyIdentifier の情報を項目毎に階層表示。なお、表示の際は以下のルールを適用する。 ・項目名は英語のまま。 ・OID は RFC の規定値に変換。
9	拡張鍵用途	ExtKeyUsage	serverAuth clientAuth codeSigning emailProtection ipsecEndSystem ipsecTunnel ipsecUser timeStamping	ExtKeyUsage の情報を項目毎に階層表示。なお、表示の際は以下のルールを適用する。 ・項目名は英語のまま。 ・OID は RFC の規定値に変換。

<メッセージの詳細>

表 6-8 証明書検証結果「無効」の場合のエラーコード一覧

#	エラーコード	内容	意味
1	100101	認証パス構築不可(101)	認証パス構築ができないこと。
2	100202	署名不正(202)	認証パスに署名が不正である証明書が含まれていること。
3	100203	失効証明書を含む(203)	認証パスに失効した証明書が含まれていること。
4	100204	ポリシー不一致(204)	認証パスにポリシーが一致しない証明書が含まれていること。
5	100205	制約違反(205)	認証パスに制約に違反している証明書が含まれていること。
6	100206	OCSP での証明書検証確認不正(206)	認証パスに OCSP での certStatus が unknown と応答される証明書が含まれていること。
7	100301	アルゴリズム拒否(301)	官職証明書検証サーバ(CVS)側で検証対象証明書の受け付けを拒否されたこと。
8	100302	アルゴリズム拒否(302)	証明書検証サーバ(CVS)側で中間証明書または TA 証明書の受け付けを拒否した。
9	100901	要求受け付け拒否(901)	官職証明書検証サーバ(CVS)側で要求の受け付けが拒否されたこと。

注：()中のコードは cert_path_status を示す。

表 6-9 証明書検証結果「検証失敗」の場合のエラーコード一覧

#	エラーコード	内容	意味
1	100902	タイムアウト(902)	要求がタイムアウトとなったこと。()中のコードは cert_path_status を示す。
2	200100	署名検証失敗	受信した OCSP レスポンスデータが改竄されていること。
3	200200	証明書が改竄または、有効期間切れ	OCSP レスポンスに付与されている電子証明書が改竄または有効期間切れである。
4	300100	接続失敗	ネットワークの問題で通信できなかった。プロキシサーバを使用している場合は、プロキシ情報の指定に誤りがある。
5	300200	拡張領域の解析に失敗	官職証明書検証サーバ(CVS)に接続していない。環境設定ファイルの CVS 接続先 URL をもう

#	エラーコード	内容	意味
			一度見直してください。
6	300300	その他のエラー発生のため証明書検証の確認不可	<ul style="list-style-type: none"> 環境設定ファイルに CVS 接続先 URL が指定されていない又は、環境設定ファイルが存在しない。 その他の内部エラーが発生。
7	300400	証明書の取得失敗	<ul style="list-style-type: none"> IC カードの PIN 入力でキャンセルされた場合。 IC カードが端末にセットされていない場合。
8	400001	OCSP Request のフォーマットエラー (1)	官職証明書検証サーバ(CVS) で例外が発生 <ul style="list-style-type: none"> 検証要求を行う官職証明書検証サーバ(CVS) の検証依頼者認証が必要であるか確認してください。 署名に使用した電子証明書が X.509 バージョン 3 の電子証明書であるかを確認してください。
9	400002	内部エラー (2)	
10	400003	一時的な回答不能(3)	
11	400005	OCSP Request への署名が必要(5)	
12	400006	クライアントが認証されていない(6)	

注：()中のコードは response_status を示す。

第8節 利用者証明書有効性確認画面

画面名	利用者証明書有効性確認画面	
概要	電子証明書（基本 / 詳細）画面にて、[有効性確認]ボタンを押下し、有効性確認を行い、有効性確認結果を表示する。	
画面レイアウト		
		
画面項目説明		
#	項目名	概要
	タイトルバー	表示テキストは「検証結果表示」とする。

有効性確認結果	<p>有効性確認結果が終了した場合、以下のメッセージを表示する。</p> <p>対象とする証明書が有効の場合 有効性確認結果：「有効」</p> <p>対象とする証明書の有効期間が切れている場合 有効性確認結果：「有効期間切れ」</p> <p>対象とする証明書が失効済みの場合 有効性確認結果：「失効済み」</p> <p>対象とする証明書のオンライン失効申請を行った場合 有効性確認結果：「失効申請受理済み」</p> <p>対象とする証明書が一時保留状態の場合 有効性確認結果：「一時保留」</p> <p>有効性確認処理が失敗した場合 有効性確認結果：「確認失敗 (xxxxxx)」 (確認失敗とは、有効性確認が行えなかった場合を示す。) 「xxxxxx」は原因を示すエラーコード。詳細は、表 6-10 表 6-10 有効性確認結果「確認失敗」のエラーコード一覧を参照。</p>
OK ボタン	利用者証明書有効性確認画面を閉じ、前画面に戻る。
閉じるボタン	利用者証明書有効性確認画面を閉じ、前画面に戻る。
バックキー	利用者証明書有効性確認画面を閉じ、前画面に戻る。

<メッセージの詳細>

表 6-10 有効性確認結果「確認失敗」のエラーコード一覧

#	エラーコード	内容	意味
1	100600	申請書の電子署名検証エラー	オンライン窓口サーバで「申請書の電子署名検証エラー」が発生した。
2	100604	対象とする証明書の認証局の電子署名検証エラー	オンライン窓口サーバで「対象とする証明書の認証局の電子署名検証エラー」が発生した。
3	100605	対象とする証明書の有効性確認エラー	オンライン窓口サーバで「対象とする証明書の有効性確認エラー」が発生した。
4	200601	申請情報取込エラー	オンライン窓口サーバで「申請情報取込エラー」が発生した。
5	200602	対象とする証明書の有効期間切れエラー	オンライン窓口サーバで「対象とする証明書の有効期間切れエラー」が発生した。

#	エラーコード	内容	意味
6	200603	受信データ形式エラー	オンライン窓口サーバで「受信データ形式エラー」が発生した。
7	200609	オンライン窓口エラー (DB)	オンライン窓口サーバで「オンライン窓口エラー(DB)」が発生した。
8	200611	オンライン窓口混雑	オンライン窓口サーバが混雑している。
9	200612	オンライン窓口エラー	オンライン窓口サーバでエラーが発生した。
10	200622	対象とする証明書と署名実施証明書の不一致エラー	オンライン窓口サーバで「対象とする証明書と署名実施証明書の不一致エラー」が発生した。
11	200652	署名実施証明書の有効期間外エラー	オンライン窓口サーバで「署名実施証明書の有効期間外エラー」が発生した。
12	200654	署名実施証明書の認証局の電子署名検証エラー	オンライン窓口サーバで「署名実施証明書の認証局の電子署名検証エラー」が発生した。
13	200655	署名実施証明書の有効性確認エラー	オンライン窓口サーバで「署名実施証明書の有効性確認エラー」が発生した。
14	200656	オンライン窓口エラー(OCSP)	オンライン窓口サーバで「オンライン窓口エラー(OCSP)」が発生した。
15	200657	署名実施証明書の失効済みエラー	オンライン窓口サーバで「署名実施証明書の失効済みエラー」が発生した。
16	200658	署名実施証明書の失効申請受理済みエラー	オンライン窓口サーバで「署名実施証明書の失効申請受理済みエラー」が発生した。
17	200663	署名実施証明書の一時保留状態エラー	オンライン窓口サーバで「署名実施証明書の一時保留状態エラー」が発生した。
18	2001xx	HTTP 通信エラー	HTTP 通信中にエラーが発生した。 1xx ~ 5xx は HTTP1.1 ステータスコード。
19	2002xx		
20	2003xx		
21	2005xx		
22	3004xx		
23	300703	ICカードがロックされている	ICカードは現在使用中。
24	300704	ICカードに接続できない	カードリーダ接続不備、またはカードが接続されていない。
25	300705	対象とする証明書の解析失敗	対象とする証明書の解析に失敗した。
26	300706	サーバ時刻の取得に失敗	サーバ時刻の取得に失敗した。

#	エラーコード	内容	意味
27	300707	オンライン窓口サーバへのアクセスエラー	オンライン窓口サーバでアクセスに失敗した。
28	300709	対象とする証明書の有効期間になっていない	対象とする証明書の有効期間開始の日付が未来の日時である。
29	400606	対象とする証明書のオンライン窓口エラー (OCSP)	オンライン窓口サーバで「対象とする証明書のオンライン窓口エラー (OCSP)」が発生した。

第9節 パスワード入力画面

画面名	パスワード入力画面	
概要	パスワードの入力を案内する。	
画面レイアウト		
画面項目説明		
#	項目名	概要
	タイトルバー	表示テキストは「パスワード入力」とする。
	パスワード入力案内メッセージ	署名用電子証明書の場合：「公的個人認証サービス 署名用パスワード(英数字6～16文字)を入力してください。」 利用者証明用電子証明書の場合：「公的個人認証サービス 利用者証明用パスワード(数字4桁)を入力してください。」
	パスワード	パスワードを入力する。
	パスワードを表示する チェックボックス	パスワードの伏字表示を切り替える。 デフォルトはチェックなしとする。

	注意メッセージ	<p>署名用電子証明書の場合：</p> <p>「 注意</p> <ul style="list-style-type: none">・本パスワードは5回連続で間違えるとロックされますのでご注意ください。 <p>ロックした場合の初期化は、市区町村窓口等やコンビニのキオスク端末（専用スマホアプリが必要）により行うことができます。詳しくはメニュー画面の「JPKI ポータルサイト閲覧」から JPKI ポータルサイトをご確認ください。」</p> <p>利用者証明用電子証明書の場合：</p> <p>「 注意</p> <ul style="list-style-type: none">・本パスワードは3回連続で間違えるとロックされますのでご注意ください。 <p>ロックした場合は市区町村窓口にてパスワード初期化を申請してください。」</p>
	OK ボタン	パスワード入力画面を閉じ、前画面に戻る。
	キャンセルボタン	パスワード入力画面を閉じ、前画面に戻る。
	終了ボタン	パスワード入力画面を閉じ、前画面に戻る。
	バックキー	パスワード入力画面を閉じ、前画面に戻る。

第 10 節 IC カードセット案内画面

画面名	IC カードセット案内画面	
概要	IC カードセットの案内を行う。	
画面レイアウト		
		
画面項目説明		
#	項目名	概要
	タイトルバー	「IC カードセット案内」を表示する。
	終了ボタン	IC カードのセットを中止し、復帰先画面に戻る。
	ICカードセット案内メッセージ	「マイナンバーカードをスマートフォンの NFC 読み取り位置にぴったりと密着させてください。」を表示する。
	ポータルサイトボタン	ブラウザ経由でポータルサイトを表示する。
	バックキー	IC カードのセットを中止し、復帰先画面に戻る。

第11節 設定ルール一覧

1 代替文字の設定ルール

() 表記ルール

1. 代替文字を"1"、それ以外を"0"で表現する。
2. スペースも1文字として捉え、ルール1を適用する。

() 表記例

項目名	設定値	代替文字使用位置の値	説明
氏名	吉田 太郎	10000	氏名の長さは5文字 1文字目の「吉」が代替文字
住所	東京都葛飾区 x x x	000100000	住所の長さは9文字 4文字目の「葛」が代替文字

は全角スペース

2 生年月日の設定ルール

() コード体系

英数字型 9桁 EYYYYMMDD

E : 年号コード 1桁 (1:明治 2:大正 3:昭和 4:平成 5:令和)

YYYY : 西暦年 4桁

MM : 月 2桁 (01~12:1月~12月 00:不明 A1:春 A2:夏 A3:秋 A4:冬)

DD : 日 2桁 (01~31:1日~31日 00:不明 A1:上旬 A2:中旬 A3:下旬)

() 表記例

例	生年月日の値	表記
通常	420030401	平成 15 年 4 月 1 日
年号のはざまの日	219261225	大正 15 年 12 月 25 日
	319261225	昭和元年 12 月 25 日
年月日不明	000000000	
月日不明	319260000	昭和元年
	31926A100	昭和元年春
日不明	319261200	昭和元年 12 月
	3192612A2	昭和元年 12 月中旬

3 性別の設定ルール

() コード体系

英数字型 1桁 X

X : 性別コード1桁 (1:男 2:女 3:不明)

禁・無断転載

公的個人認証サービス

利用者クライアントソフト API 仕様書
【Android インテント 編】

第 1.7 版

(注意事項)

利用者クライアントソフトの著作権は、総務省、地方公共団体情報システム機構が保有しており、国際著作権条約及び日本国の著作権関連法令によって保護されています。

利用者クライアントソフトの利用に当たっては、次に掲げる行為を禁止します。

- (1) 利用者クライアントソフトを電子署名に係る地方公共団体情報システム機構の認証業務に関する法律において制限されている電子証明書の用途で利用すること。
- (2) 利用者クライアントソフトに対し、総務省、地方公共団体情報システム機構に許可なく改造等を行うこと。

総務省、地方公共団体情報システム機構は、利用者が利用者クライアントソフトを利用したことにより発生した利用者の損害及び利用者が第三者に与えた損害について、一切の責任を負いません。

商標については次の通りです。

- (1) Android は、Google LLC の米国およびその他の国における登録商標です。
- (2) その他、記載されている会社名、製品名等は、各社の登録商標または商標です。