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1.3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			
		Java9 対応に伴い、以下を修正。			
		・ 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			
1.1版	亚世 20 年 7 日 24 日	Android 6.0.1、7.0 を追加。			
1.1 h/X	平成 29 年 7 月 31 日 	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。			
		Android SDK(26)対応に伴い、以下を修正。			
		第3章 第1節 表3-1の【PC 接続の場合】に			
1.2版	平成 30 年 8 月 31 日	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。			
		新元号対応に伴い、以下を修正。			
		・ 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。			
1.3版	 令和元年 5 月 1 日	・ 第6章 第11節 2生年月日の設定ルールに新			
1.5 діх		元号を追加。			
		・その他、記載の見直しを実施。			
		旧氏対応に伴い、以下を修正。			
		・ 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。			

- 目次 -

第1	1章 はじめに	. 1
穿	91節 用語の定義	2
第 2	2章 ドキュメント体系	. 3
第3	3 章 動作環境	. 5
穿	自1節 Android 端末	5
穿	92節 前提条件	6
第4	4章 機能仕様	. 7
穿	81節 ソフトウェア構成図	7
穿	82節 実現可能な機能の一覧	8
第5	5章 インテント仕様	. 9
穿	§1節 インテントインターフェースの概念図	9
	1 送信データ	1 0
	2 受信データ	1 1
穿	第2節 拡張データに格納する項目一覧	1 3
	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)	1 3
	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)	1 4
	3 カード AP ライブラリ(署名用の場合)	1 5
	4 カード AP ライブラリ(利用者証明用の場合)	2 1
	5 個人認証サービス AP(署名用、利用者証明用共通)	2 5
笋	93節 コーリングシーケンス	3 2
	1 初期処理	3 2
	2 終了処理	3 4
	3 カード AP ライブラリ(署名用の場合)	3 6
	4 カード AP ライブラリ(利用者証明用の場合)	4 6
	5 個人認証サービス AP(署名用、利用者証明用共通)	5 2
第6	5章 画面仕様	6 0
穿	91節 画面一覧	6 0
穿	92節 署名用電子証明書基本画面	6 2
穿	93節 利用者証明用電子証明書基本画面	6 5
穿	94節 認証局の自己署名証明書基本画面	6 8
穿	95節 その他の電子証明書基本画面	7 1
穿	96節 詳細画面	7 4
穿	87節 官職証明書・職責証明書検証画面	8 0
穿	88節 利用者証明書有効性確認画面	8 3
穿	9 節 パスワード入力画面	8 7
笋	910節 IC カードセット案内画面	8 8
爭	911節 設定ルール一覧	8 9

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、日本工業規格 X560-1 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			
5	認証局の自己	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。			
	署名証明書	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			
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。

利用者クライアントソフト 機能概要説明書 利用者クライアントソフトの機能概要について説明しています。 API 仕様書 カード AP ライブラリ CryptoAPI 編 カード AP ライブラリ(CryptoAPI)の API 仕様について説明しています。 API 仕様書 カード AP ライブラリ PKCS#11 編 カード AP ライブラリ(PKCS#11)の API 仕様について説明しています。 API 仕様書 カード AP ライブラリ Java インターフェース編 カード AP ライブラリ(Java インターフェース)の API 仕様について説明しています。 JavaDoc JPKICryptJNI カード AP ライブラリ(Java インターフェース(住基カード署名用))の JavaDoc です。 JavaDoc JPKICryptJNI(64bit) カード AP ライブラリ(Java インターフェース(住基カード署名用) 64bit)の JavaDoc です。 JavaDoc JPKICryptSignJNI カード AP ライブラリ(Java インターフェース(個人番号カード署名用))の JavaDoc です。 JavaDoc JPKICryptSignJNI(64bit) カード AP ライブラリ(Java インターフェース(個人番号カード署名用) 64bit)の JavaDoc です。 JavaDoc JPKICryptAuthJNI カード AP ライブラリ(Java インターフェース(個人番号カード利用者証明用))の JavaDoc です。 JavaDoc JPKICryptAuthJNI(64bit) カード AP ライプラリ(Java インターフェース(個人番号カード利用者証明用)64bit)の JavaDoc です。 API 仕様書 個人認証サービス AP C言語インターフェース編 個人認証サービス AP(C言語インターフェース)の API 仕様について説明しています。 API 仕様書 個人認証サービス AP Java インターフェース編 個人認証サービス AP(Java インターフェース)の API 仕様について説明しています。 JavaDoc JPKIUserCertService 個人認証サービス AP(Java インターフェース)の JavaDoc です。 API 仕様書 Mac OS X C 言語インターフェース編 Mac OS X の C インターフェースの API 仕様について説明しています。 Jar ファイル利用手引き 利用者クライアントソフトの Jar ファイルを利用する方法について説明しています。

図 2-1 ドキュメント体系図(PC版)

|利用者クライアントソフト 機能概要説明書(Android 版)

Android 版 JPKI 利用者ソフトの機能概要について説明しています。

API 仕様書 Android インテント編

インテントによる外部インターフェース仕様について説明しています。

図 2-2 ドキュメント体系図(Android 版)

第3章 動作環境

第1節 Android 端末

Android 版 JPKI 利用者ソフト Ver1.1 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		
	【PC 接続の場合】Android 4.3、5.1、6.0.1、7.0 または8.0 を搭載		
OS	していること。		
	【Android 単体で利用する場合】Android 5.1、6.0.1、7.0 または 8.0		
	を搭載していること。		
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。

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 Ver1.1 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	AP	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。

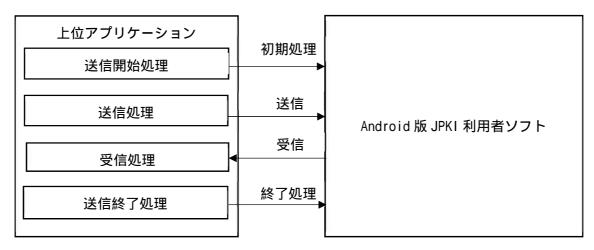


図 5-1 インテントインターフェースの概念図

1 送信データ

図 5-1 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	Intent.ACTION_SEND を設定すること。		
	JPKI 利用者ソフトの	以下のように設定すること。		
	パッケージ名とクラ	intent.setClassName(
	ス名の設定	<jpki)="" 利用者ソフトのパッケージ名(="">,</jpki>		
		"jp.go.jpki.mobile.intent.JPKIIntentActivity")		
		;		
		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:"jp.go.jpki.mobile.utility"		
		署名検証者向け:		
		"jp.go.jpki.mobile.utility.dev"		
	拡張データ	intent.putExtra 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 onActivityResult 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		
requestCode	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。		
resultCode	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を参照すること。		
data	拡張データ	intent.putExtra 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	command_type	int	0x01004001	コマンドタイプ	
タ					
受信デー	command_type	int	0x01004001	コマンドタイプ	
タ	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	command_type	int	0x01004002	コマンドタイプ		
タ						
受信デー	command_type	int	0x01004002	コマンドタイプ		
タ	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	command_type	int	0x01002001	コマンドタイプ			
タ	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	処理の流れは以下となる。						
	移する。 ・ IC カードから署	署名用認証 原	司の自己署名証[カードセット案内画面へ画面遷 明書を取得する。 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	3署名用電·	子証明書(DER 形	(式)を取得			
	キー	一型值内容					
送信デー	command_type	int	0x01002002	コマンドタイプ			
タ							
受信デー	command_type	ommand_type int 0x01002002 コマンドタイプ					
タ	p_cert	byte[] 秘密鍵に対応する署名用電子証					
		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	処理の流れは以口	ことなる。			
	・ 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カードかり	ら秘密鍵に	対応する署名用	電子証明書を取得する。	
	・ 上位アプリク	アーション	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	直を生成				
	キー	型	値内容			
送信デー	command_type	int	0x01002003	コマンドタイプ		
タ	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	command_type	int	0x01002003	コマンドタイプ		
タ	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。

- ・ 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。

ハッシュ値への OID 付与は以下のバイナリ形式で行う。

・ ハッシュアルゴリズムが SHA1 の場合(データサイズは 35Byte)

0x30, 0x21,

0x30, 0x09,

0x06, 0x05, 0x2b, 0x0e, 0x03, 0x02, 0x1a, 0x05, 0x00,

0x04, 0x14,

SHA1 ハッシュ値(20Byte)

・ ハッシュアルゴリズムが SHA256 の場合(データサイズは 51Byte)

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	キー 型 値 内容				
送信デー	command_type	int	0x01002004	コマンドタイプ		
タ	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	command_type	int	コマンドタイプ			
タ	signature	byte[]		署名值		
	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				
		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。 ・ 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。
	 ハッシュ値への OID 付与に関する注意事項は以下の通り。 ・ 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					
	キー	型值内容				
送信デー	command_type	int	0x01002005	コマンドタイプ		
タ	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	command_type	int	0x01002005	コマンドタイプ		
タ	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	処理の流れは以	下となる。			
	・ 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。				
	・ 上位アプリク	ケーション	に検証結果を返	す 。	
	署名値に関する	主意事項は	以下の通り。		
	・ 上位アプリク	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(3)備考「ハッシュ値への OID	
	付与」を参照	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	command_type	int	0x01002006	コマンドタイプ		
タ	certificate	byte[]		証明書データ(DER 形式)		
	hash	byte[]		ダイジェストデータ		
	signature	byte[]		署名值		
	alg_id	int		ハッシュアルゴリズムの指定		
				SHA1:0		
				SHA256:1		
受信デー	command_type	int	0x01002006	コマンドタイプ		
タ	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	・ 上位アプリ ダイジェストテ ・ OID を付与 プリケーシ ること。(3 参照のこと ・ OID を付与	ュ値、署名値、公開鍵を使用して、署名値を検証する。 プリケーションに検証結果を返す。 トデータと署名値に関する注意事項は以下の通り。 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。(第5章 第2節 3(3)備考「ハッシュ値への OID 付与」を				
	ータを入力すること。					
		ダイジェストデータ				
			OID	あり	OIDなし	
	署名值	OIDあり	(OK	NG	
		OID なし	, 1	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

		白己署名記						
		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	command_type	int	0x01001001	コマンドタイプ				
9	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	移する。 ・ IC カードから利用	されていない 計者証明用	認証局の自己署	フードセット案内画面へ画面遷 名証明書を取得する。 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	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	command_type	int	0x01001002	コマンドタイプ	
タ					
受信デー	command_type	int	0x01001002	コマンドタイプ	
タ	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	

		1	ı	
				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。		
	・ IC カードに接続る	されていな	い場合は、IC た]ードセット案内画面へ画面遷
	移する。			
	・ IC カードから秘密	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	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	と 成			
	キー	型	値	内容	
送信デー	command_type	int	0x01001003	コマンドタイプ	
タ	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	command_type	int	0x01001003	コマンドタイプ	
タ	signature	byte[]		署名值	
	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	ڼ る。			
	・ IC カードに接続で	されていた	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。

ハッシュ値への OID 付与は以下のバイナリ形式で行う。

・ 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(ダイシ			
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	command_type	int	0x01001004	コマンドタイプ
タ	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	command_type	int	0x01001004	コマンドタイプ
タ	signature	byte[]		署名值
	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			
		されていた	IN場合は、IC	カードセット案内画面へ画面遷
	移する。	- 10-	L 181-45/4-2-	
	-) C \ 10 7	リードに格納さ	れた利用者秘密鍵を使用して署名
	値を生成する。	> . = =	マタケン	
	・ 上位アプリケーシ 	/ヨンに書	1台他を返り。	
	 ハッシュ値への 0ID f	t 노ic 悶a	ス注音車頂け!	リエの通り
			マエ心中点は	グ 1. ひにり。

- ・ 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			
	‡ -	型	値	内容
送信デー	command_type	int	0x01003001	コマンドタイプ
タ	cert	byte[]		証明書データ(DER 形式)
受信デー	command_type	int	0x01003001	コマンドタイプ
タ	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。			
	・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	command_type	int	0x01003002	コマンドタイプ
タ	cert	byte[]		証明書データ(DER 形式)
受信デー	command_type	int	0x01003002	コマンドタイプ
タ	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
	Hame	July	C石 (大括弧「[]」内は旧氏を
			示す。括弧「()」内は通
			称を示す。)
	substitute_characte	String	住所の代替文字
	r_of_address		(第6章 第11節 設定ル
			ール一覧参照)
	substitute_characte	String	氏名の代替文字
	r_of_name		(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	<u>, , , , , , , , , , , , , , , , , , , </u>	
	・ 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	
	+ -	型	値	内容	
送信デー	command_type	int	0x01003003	コマンドタイプ	
タ	cert	byte[]		証明書データ(DER 形式)	
受信デー	command_type	int	0x01003003	コマンドタイプ	
タ	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検証結果コードの	
				值一覧参照)	
	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	処理の流れは以下となる	 პ。			
	・ IC カードに接続さ	れていない場合は、I	C 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	行い、結果を取得する) ₀		
	・ 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。
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。

値	意味
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
	<i>た</i> 。
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	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		
	‡ -	型	値	内容
送信デー	command_type	int	0x01003004	コマンドタイプ
タ	cert	byte[]		証明書データ(DER 形式)
受信デー	command_type	int	0x01003004	コマンドタイプ
タ	code	int		有効性確認結果
				(表 5-7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	処理の流れは以下となる。
	・ 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。	
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。	

値	意味
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	HTTP1.1 ステータスコード(100~500 番台)。
(200 を除く)	

(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	command_type	int	0x01003005	コマンドタイプ
タ				
受信デー	command_type	int	0x01003005	コマンドタイプ
タ	id	int		IC カード種別
				(表 5-8 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	処理の流れは以下となる。			
	・ 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。			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	引数
成	・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

Intent インスタンスの生	-
	引数
成	・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"
	31 312X. Communica_1)Po
	313X. command_1/pc
data.getBooleanExtra	結果を取得(boolean 型)
data.getBooleanExtra	T
data.getBooleanExtra	結果を取得(boolean 型)
data.getBooleanExtra	結果を取得(boolean 型) 引数
data.getBooleanExtra	結果を取得(boolean 型) 引数 ・第一引数: "result"
data.getBooleanExtra data.getIntExtra	結果を取得(boolean 型) 引数 ・第一引数: "result"

data.getIntExtra	詳細コードを取得(int 型)

取得した値が 0000 の場合は正常に処理された

・第一引数: "err_code"

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

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 初期処理参照	
<u> </u>		
Intent インスタンスの生	引数	
成	・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)	
	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 終了処理参照	
---------------	--

初期処理	1 初期処理参照
Intent インスタンスの生	引数
成	・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)

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
	ANN OPPIERS COOK OF WILLIAM PROCESSION
data.getIntExtra	詳細コードを取得(int型)
data.getiiitExtra	計画コーで取得(III 主) 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
	1
終了処理	2 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 初期処理参照
Intent インスタンスの生	引数
成	・第一引数: Intent.ACTION_SEND
774	313 313XV 11100110111011 <u>-</u> 02115
intent.setClassName	JPKI 利用者ソフトのパッケージ名とクラス名の設定
Tittett . Se to lassitaile	引数
	「」
	1, 5, 1,
	・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
	N-11xv. v) (v[] =

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"

はata.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

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 初期処理参照

Intent インスタンスの生 引数 成 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"
	Jp.go. jpki modific. intolition ki intolitikoti vity
intent mutfutus	transis - A catasta - A ca
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: 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
	・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)
mtomtiput2xtru	引数
	コース ・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	・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・
, , , , , , , , , , , , , , , , , , , ,	
data_getIntExtra	コマンドタイプを取得
data.getIntExtra	コマンドタイプを取得
data.getIntExtra	引数
data.getIntExtra	
	引数 ・第一引数: "command_type"
data.getIntExtra data.getByteArrayExtra	引数 ・第一引数: "command_type" 署名値の取得
	引数 ・第一引数: "command_type" 署名値の取得 引数
	引数 ・第一引数: "command_type" 署名値の取得
	引数 ・第一引数: "command_type" 署名値の取得 引数
	引数 ・第一引数: "command_type" 署名値の取得 引数
data.getByteArrayExtra	引数 ・第一引数: "command_type" 署名値の取得 引数 ・第一引数: "signature"

・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 初期処理参照
Intent インスタンスの生	引数
成	・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"
	11 - 30 - 11
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
	(0),(1) (4)(1) (2 0)

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 初期処理参照
------	----------

Intent インスタンスの生	引数
成	・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
	The state of the s
	引数

・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)」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)」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)」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。)

(10) 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 初期処理参照
Intent インスタンスの生	引数
成	・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)
	1. 45 5 6 # # # A B
	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
data.getiiitExtra	引数
	・第一引数: "command_type"
	・第 引致. 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
data.getbytehilayLitla	利用自証明用認証周の自己省日証明首(DLN ルム)の取得 引数
	・第一引数: "ca_cert"
	NJ JIW. 00_0011
data.getIntExtra	エラーコードを取得(int 型)
aata gotintektia	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 初期処理参照

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

intent.setClassName	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
	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: 0x01001002

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"		71344
コラーコードを取得(int型) 引数	<u>'</u>	
引数		・第一引数: "p_cert"
引数		
引数	data.getIntExtra	エラーコードを取得(int 型)
・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 詳細コードを取得(int型) 引数 ・第一引数: "detail_code" 取得した値が0000000のの場合は正常に処理された 終了処理		` <i>'</i>
取得した値が 0000 の場合は正常に処理された 詳細コードを取得(int型) 引数		
data.getIntExtra		
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終了処理参照		取得した値が 0000 の場合は止常に処理された
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 初期処理参照 Intent インスタンスの生 引数 ・第一引数: Intent.ACTION_SEND 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.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	data.getIntExtra	詳細コードを取得(int 型)
取得した値が 00000000 の場合は正常に処理された 終了処理		引数
取得した値が 00000000 の場合は正常に処理された 終了処理		 ・第一引数: "detail code"
終了処理 (3) 署名生成処理(署名対象データを渡す場合) 初期処理 1 初期処理参照 Intent インスタンスの生成 成 リアに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: "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		_
(3) 署名生成処理(署名対象データを渡す場合) 初期処理 Intent インスタンスの生 成 ・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		以待りた他が 00000000 00場合は正常に延遅されいた
(3) 署名生成処理(署名対象データを渡す場合) 初期処理 Intent インスタンスの生 成 ・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		
Intent インスタンスの生成	終了処理	2 終了処理参照
Intent インスタンスの生成		
Intent インスタンスの生成	(3)署名生成処理(署	畳名対象データを渡す場合)
Intent インスタンスの生成 ・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		
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	[7]州]延生	「関剤及注シ派
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		
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	Intent インスタンスの生	引数
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	成	・第一引数: Intent.ACTION_SEND
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		
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	intent.setClassName	JPKI 利用者ソフトのパッケージ名とクラス名の設定
・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		
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		
 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 		"jp.go.jpki.mobile.intent.JPKIIntentActivity"
 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 		
 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 	intent.putExtra	拡張データの設定(コマンドタイプの設定)
・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		
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		*
(1),(2)で取得できるデータ 引数		
引数	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 "		・第一引数: "message"
・第二引数: byte[]型		
为— JI		

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"

はata.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 初期処理参照

Intent インスタンスの生 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: 0x01001004
	为二月效: 0,01001004
intent nutEvtro	抗運ニータの訊字(ダイジェフト / ハッシュ 値)の訊字)
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
	・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
	・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	
,	,
data.getIntExtra	コマンドタイプを取得
and go thing and	引数
	・第一引数: "command_type"
	AJ JIXX. Communia_type
data.getByteArrayExtra	署名値の取得
data.getbyteAfrayExtfa	
	引数 · 第一引数· " a i gnatura "
	・第一引数: "signature"
	I»+ III/II / · · III
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)」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)」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)」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

data.getIntExtra

初期処理	1 初期処理参照
•	
Intent インスタンスの生	引数
成	・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"
	· · · · · · · · · · · · · · · · · · ·

エラーコードを取得(int型)

・第一引数: "err_code"

引数

	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 終了処理参照
(2) 基本4情報取得	
初期処理	1 初期処理参照
Intent インスタンスの生	引数
成	・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

	T
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
autungereri ing_mira	引数
	・第一引数: "date_of_birth"
	33 31xx. dato_01_511tti
data gatetringEvtro	性別を取得
data.getStringExtra	当数
	・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
	からしたにの 0000 07%日は正市にだ注じ10C
data.getIntExtra	詳細コードを取得(int型)
uata.yotiiitLxtta	計画コートを取得(IIII 室) 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
T	1
終了処理	2 終了処理参照

(3) 官職証明書検証

(3) 旨概証明書快証	
初期処理	1 初期処理参照
Intent インスタンスの生	引数
成	・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)	
· ,	
data.getIntExtra	コマンドタイプを取得
J	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
3.7	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
3	引数
	・第一引数: "cert_path_status"
	212 212XX
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
aataigotintextia	引数
	・第一引数: "response_status"
	JIXX. 100ponoo_otatao
data.getIntExtra	エラーコードを取得(int 型)
data.gotimtExtra	3数 3 3 3 3 3 3 3 3 3
	・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
	状帯のに過か 0000 の場合は正常に延延された
data.getIntExtra	 詳細コードを取得(int 型)
data.getiiitExtra	計画コートを取得(III(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 終了処理参照
(4) 自分の電子証明	明書の有効性確認
初期処理	1 初期処理参照

(4) 自分の電子証明書の有効性確認	
初期処理	1 初期処理参照
•	
Intent インスタンスの生	引数
成	・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

	T
	・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	有効性確認結果を取得
	引数
	・第一引数: " code "
data.getStringExtra	有効性確認中に発生した Exception のメッセージを取得
	引数
	・第一引数: "exception"
	·
data.getStringExtra	確認結果メッセージを取得
January Grand	引数
	・第一引数: "confirm_msg"
	5
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
	MA CALLER COCCOSC OF WALKER PROCESSES FOR
終了処理	2 終了処理参照
心」处注	4 於「たはず出

(5) ICカード種別取得

Intent インスタンスの生 成 ・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.utility" ・第二引数: "jp.go.jpki.mobile.utility" 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: Ox01003005 StartActivityForResult Android 版 JPKI 利用者ソフトの起動と intent 送信 引数 ・第一引数: 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"	初期処理	1 初期処理参照
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: Ox01003005 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"	1370372-1	· 10/10~-10/11
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: Ox01003005 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"	Intent インスタンスの生	引数
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"		
引数	17%	33 JIXX. HITCHT./NOTTON_CEND
引数	intent setClassName	IPKI 利田老ソフトのパッケージタとクラフタの設定
・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"	Titteitt. Se to lassiname	
・第二引数: "jp.go.jpki.mobile.intent.JPKIIntentActivity" intent.putExtra		
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.getIntExtra ICカード token 情報を取得 引数 ・第一引数: "token"		
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"		
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"]p.go.]p.t
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"	intent.putExtra	拡張データの設定(コマンドタイプの設定)
・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"		· · · · · · · · · · · · · · · · · · ·
*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"		
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"		
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"		
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"	startActivityForResult	Android 版 JPKI 利用者ソフトの起動と intent 送信
・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 ICカード token 情報を取得引数。・第一引数: "token"	·	引数
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"		・第一引数: intent
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"		・第二引数:任意のリクエストコード(int 型)
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 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"	Android 版 JPKI 利用者ソフ	トからの情報を onActivityResult(int requestCode, int
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"	resultCode, Intent data)	で受け取る
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"		
・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)	data.getIntExtra	コマンドタイプを取得
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"		引数
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)		・第一引数: "command_type"
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)		
・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)	data.getIntExtra	IC カード種別を取得
data.getStringExtra IC カード token 情報を取得 引数 ・第一引数: "token"		引数
引数 ・第一引数: "token" data.getIntExtra エラーコードを取得(int型)		・第一引数: "id"
引数 ・第一引数: "token" data.getIntExtra エラーコードを取得(int型)		
・第一引数: "token" data.getIntExtra エラーコードを取得(int型)	data.getStringExtra	ICカード token 情報を取得
data.getIntExtra エラーコードを取得(int 型)		引数
		・第一引数: "token"
引数	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	基本画面	署名用電子	署名用電子証明書の基本情報を表示する。
	機能		証明書	[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	利用者証明用電子証明書の基本情報を表示する。
			用電子証明	[有効性確認]ボタンの押下により、自分の電子証
			書	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	認証局の自己署名証明書(ルート認証局の自己署
			己署名証明	名証明書,リンク証明書,下位認証局の自己署名
			書	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	上記以外の証明書(官職証明書,職責証明書,そ
			子証明書	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			・職責証明書	その他の電子証明書(基本/詳細)画面にて、[証
		検証画面		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	署名用電子証明書(基本/詳細)画面または利用
		画面	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



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)を参照。

		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	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	氏名の代替文字を表示する。代替文字がない場合は、表示しな
(氏名) 1	ll _°
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	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	住所の代替文字を表示する。代替文字がない場合は、表示しな
(住所) 1	ι I _o
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	電子証明書の有効期間の満了日を和暦で表示する。
日	
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	[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
ン	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	「有効性確認結果」を表示する。
トル	
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(xxxxxxx)」
	「xxxxxxx」は原因を示すエラーコード。詳細は、表 6 - 1 0 有
ᆂᇸᄣᅓᅒᅶ	対性確認結果「確認失敗」のエラーコード一覧を参照。
有効性確認ボタ	[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
ン	ービス AP の自分の電子証明書の有効性確認機能を呼び出し、
	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 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	月書の項目名	表示方法
"	7,11	上位項目名	項目名	230374
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
8	有効期間 の満了日		notAfter	年 MM 月 DD 日」(GGは元号、 時分秒は表示せず。)
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 - 1
	0 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を呼
	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	証印	明書の項目名	表示方法
	ZHH	上位項目名	項目名	100012
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
3	有効期間 の満了日		notAfter	年 MM 月 DD 日」(GGは元号、 時分秒は表示せず。)
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

	認証局の自己署名証明書基本画面
概要認	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	2377774
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「G G YY
3	有効期間 の満了日		notAfter	年 MM 月 DD 日」(GGは元号、 時分秒は表示せず。)
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。

API 仕様書【Android インテント編】第 1.3 版

5	フィンガ	-	-	電子証明書のハッシュ値を計
	ープリン			算して表示。ハッシュ関数は
	F			「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		
佩女	する。		

画面レイアウト * 10 **4**× 10 **1** 電子証明書表示 電子証明書 基本情報 詳細情報 主体者 CN=テスト用証明書 0 1 A OU=テスト省 O=日本国政府 C=JP 発行年月日 平成25年4月30日 有効期間の満了日 平成28年4月29日 発行者 OU=デモ官職認証局 O=日本国政府 C=JP 証明書検証結果 電子証明書の検証は行われていません。 証明書検証 閉じる V

表示項目と証明書領域の対応は、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。		
		・ 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	[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		
ン	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	「証明書検証結果」を表示する。		
ル			
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)」		
	「xxxxxxx」は原因を示すエラーコード。詳細は、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を呼び出し、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	明書の項目名	表示方法
	7,11	上位項目名	項目名	
1	主体者	SubjectAltName	CountryName	SubjectAltName の DN を全て
		または	OrganizationName	表示。
		Subject	OrganizationalUnitName	DN は属性毎に改行して表示
			CommonName	し、「(属性名の略語) = (設定
			例)官職証明書の場合	値)」にて表記。
				subjectAltName に記述がな
				い場合は subject を表示。
2	発行年月	Validity	notBefore	設定値を和暦(日本標準時)に
	日			変換して表示。書式は「GGYY
3	有効期間		notAfter	年 MM 月 DD 日」(G G は元号、
	の満了日		•	時分秒は表示せず。)
4	発行者	IssuerAltName	CountryName	IssuerAltName の DN を全て表 -
		または	OrganizationName	示。
		Issuer	OrganizationalUnitName	DN は属性毎に改行して表示
			OrganizationalUnitName 例)官職証明書の場合	し、「(属性名の略語) = (設定 値)」にて表記。
			1711 白楓証明音の物口	issuerAltName に記述がない
				場合は issuer を表示。
5	フィンガ	-	-	認証局の自己署名証明書の場
	ープリン			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項目名と証明書基本領域との対応、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	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。
		・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	電子証明書のハッシュ値を計算して表示する。
		ハッシュ関数は「sha1」または「sha256」を使用する。
	証明書内容表示領域	電子証明書内の記載事項を表示する。
		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
		域との対応と表 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。
	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	表示テキストは以下の通りとする。
	ル	・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 /	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	
証明書検証ボタン	確認機能については、第2節 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		丰二七汁
#	タロロ タロロ	上位項目名	項目名	表示方法
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	signature	algorithm	algorithmのOIDをRFCの規定値に
	ズム		parameters	変換した値。
4	発行者	issuer	countryName	DN を属性毎に改行。各行は「(属性
			organizationName	の略語)=(設定値)」にて表記。
			organizationalUni	<属性の略語>
			tName	rcountryName」
			等	С
				^r organizationName」
				0
				rorganizationalUnitName」
				OU
5	発行年月日	Validity	notBefore	設定値を西暦(日本標準時)で表
6	有効期間の満		notAfter	示。書式は「YYYY年MM月DD日hh
	了日			時 mm 分 ss 秒 」。

	#D.A	証明書の項目名		キーナバ
#	項目名	上位項目名	項目名	表示方法
7	主体者 4	subject	countryName	DN を属性毎に改行。各行は「(属性
			localityName	の略語)=(設定値)」にて表記。
			commonName	<属性の略語>
			等	rcountryName」
				С
				「localityName」
				L
				r 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	subject	commonName	commonName から発行申請送信時刻を
	時刻 5		のみが表示対象	抜き出して表示。
9	受付端末識別			commonName から受付端末識別記号を
	記号 5			抜き出して表示。
10	発行申請送信	subject	commonName	commonName から発行申請送信時刻を
	時刻 6		のみが表示対象	抜き出して表示。
11	シーケンス番			commonName からシーケンス番号を抜
	号 6			き出して表示。
12	受付端末識別			commonName から受付端末識別記号を
	記号 6			抜き出して表示。
13	ランダム文字	subject	commonName	commonName からランダム文字列を抜
	列 7		のみが表示対象	き出して表示。
14	受付端末識別			commonName から受付端末識別記号を
	記号 7			抜き出して表示。
15	主体者の公開	subjectPubl	algorithm	SubjectPublicKey の設定値(16進
	鍵情報	icKeyInfo	subjectPublicKey	数)をそのまま表示。
16	発行者ユニー	issuerUniqueID		設定値をそのまま表示。
	ク識別子			
17	主体者ユニー	subjectUnique	eID	設定値をそのまま表示。
	ク識別子			

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		書の項目名	表示方法
#	坦日石	上位項目名	項目名	48小刀压
1	認証局鍵識別	AuthorityKe	keyldentifier	AuthorityKeyIdentifier の情報を項
	子	yldentifier	authorityCertIssu	目毎に階層表示。なお、表示の際は
			er	以下のルールを適用する。
			authorityCertSeri	・項目名は英語のまま。
			alNumber	・OIDは RFC の規定値に変換。
				・DN は属性毎に改行。
2	鍵用途	KeyUsage	digitalSignature	KeyUsage の情報を項目毎に階層表
			nonRepudiation	示。なお、表示の際は以下のルール
			keyEncipherment	を適用する。
			dataEncipherment	・項目名は英語のまま。
			keyAgreement	・OIDはRFCの規定値に変換。
			keyCertSign	・KeyUsage のビット列は、鍵用途の
			cRLSign	英語名に変換してカンマ区切り表
			encipherOnly	示。
	/- +- /\) ++ /-	• • • • • • • • • • • • • • • • • • • •	decipherOnly	
3	主体者代替名	SubjectAltN	countryName organizationName	SubjectAltNameの情報を項目毎に階
		ame	organizationalUni	層表示。なお、表示の際は以下のル
			tName	ールを適用する。
			等	・項目名は英語のまま。
	びに ナル 井 ク		count ryNama	・OID は RFC の規定値に変換。
4	発行者代替名	issuerAltNa	countryName organizationName	issuerAltNameの情報を項目毎に階
		me	organizationalUni	層表示。なお、表示の際は以下のル
			tName	
			等	│・項目名は英語のまま。 │・OID は RFC の規定値に変換。
5	基本制約	BasicConstr	capathl ancapatra:	・UID は RFC の規定値に変換。 BasicConstrains の情報を項目毎に
	空 中间制	ains	capathLenConstrai nt	階層表示。なお、表示の際は以下の
		uillo	111	加一ルを適用する。
				ルールを週用する。 ・項目名は英語のまま。
				・OIDはRFCの規定値に変換。
6	 CRL 分配点	CRLDistribu	countryName	CRLDistributionPointsの情報を項
	OVE 기비깩	tionPoints	organizationName	日毎に階層表示。なお、表示の際は
		CIOIII OTITES	organizationalUni tName 等	以下のルールを適用する。
				・項目名は英語のまま。
			্য	・OIDはRFCの規定値に変換。
				・DNは属性毎に改行。
				DN 1の1年1年11年12月11日

#	項目名	証明	書の項目名	表示方法
#		上位項目名	項目名	AVIVIJIA
7	証明書ポリシ	Certificate	policyldentifier	CertificatePolicies の情報を項目
	_	Policies	policyQualifiers	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	SubjectKeyl	keyldentifier	SubjectKeyIdentifier の情報を項目
	子	dentifier		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	ExtKeyUsage の情報を項目毎に階層
			clientAuth	表示。なお、表示の際は以下のルー
			codeSigning	ルを適用する。
			emailProtection	・項目名は英語のまま。
			ipsecEndSystem	・OIDはRFCの規定値に変換。
			ipsecTunnel	
			ipsecUser	
			timeStamping	

第7節 官職証明書・職責証明書検証画面

画面名	官職証明書・職責証明書検証画面
概要	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、
佩女	証明書検証を行い、証明書検証結果を表示する。

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。
	証明書検証結果	証明書検証が終了した場合、以下のメッセージを表示する。
		証明書が有効の場合:証明書検証結果「有効」
		証明書が無効の場合:証明書検証結果「無効(xxxxxx)」
		(無効の場合とは、証明書検証結果が失敗した場合を示す。)
		証明書検証処理が失敗した場合
		: 証明書検証結果「検証失敗(xxxxxx)」
		(検証失敗とは、証明書検証が行えなかった場合を示す。)
		「xxxxxxx」は原因を示すエラーコード。詳細は、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
		証明書検証結果「検証失敗」の場合のエラーコード一覧、表
		6 - 1 0 有効性確認結果「確認失敗」のエラーコード一覧を参
		照。
	OKボタン	官職証明書・職責証明書検証画面を閉じる。
	閉じるボタン	官職証明書・職責証明書検証画面を閉じる。
	バックキー	官職証明書・職責証明書検証画面を閉じ、前画面に戻る。

<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	要求受付け拒否(901)	官職証明書検証サーバ(CVS)側で要求の受け付が拒否されたこと。

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	OCSP レスポンスに付与されている電子証明書
		効期間切れ	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
			一度見直してください。

#	エラー コード	内容	意味
6	300300	その他のエラー発生のため証明書検証の確認不可	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	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 ・ 検証要求を行う官職証明書検証サーバ
9	400002	内部エラー (2)	(CVS)の検証依頼者認証が必要であるか確
10	400003	一時的な回答不能(3)	認してください。
11	400005	OCSP Request への署名が 必要(5)	・ 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
12	400006	クライアントが認証され ていない(6)	ださい。

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
	144.55	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
			検証結果表示 ★ 10 ● *
			有効性確認結果:「有効」
			ок
#	項	 目名	概要
	タイトルバ		表示テキストは「検証結果表示」とする。
		<u>·</u>	The state of the s

有効性確認結果	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
	వ 。
	対象とする証明書が有効の場合
	有効性確認結果:「有効」
	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」
	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」
	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」
	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」
	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(xxxxxxx)」
	(確認失敗とは、有効性確認が行えなかった場合を示す。)
	「xxxxxxx」は原因を示すエラーコード。詳細は、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。
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
			ロエラー(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	オンライン窓口サーバで「対象とする証
		明書の不一致エラ ー 	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	オンライン窓口サーバで「署名実施証明
		ラー	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	オンライン窓口サーバで「署名実施証明
		署名検証エラー 	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	オンライン窓口サーバで「署名実施証明
		ラー	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
			ロエラー(OCSP)」が発生した。
15	200657	署名実施証明書の失効済みエラ	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
		<i>角の</i> エノー 	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。
19	2002xx		1xx~5xx は HTTP1.1 ステータスコード。
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



第10節 ICカードセット案内画面

し 即 し い	ートセット	·朱八四山		
画面名	IC カードセット案内画面			
概要	IC カードセ	IC カードセットの案内を行う。		
		画面レイアウト		
		□ 中		
	画面項目説明			
# Ij	[目名]	概要		
# Ij	到目名	画面項目説明		

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	「IC カードをセットしてください。」を表示する。		
	内メッセージ			
	ポータルサイトボ	ブラウザ経由でポータルサイトを表示する。		
	タン			
	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xxx	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	00000000	
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1.3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 Inc.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。