



目次

- 改訂情報
- はじめに
 - 本書の目的
 - 対象読者
 - 本書の構成
- APIリスト
 - APIリストについて
 - JavaEE開発モデル
 - スクリプト開発モデル
- プログラミング
 - 動作概念
 - APIの種類と性質
 - プログラム開発における注意点
 - 体験版ライセンスにおける注意点
- チュートリアル
 - 前提条件
 - 用語解説
 - 環境
 - サンプルプログラムの場所（すべての機能）
 - プログラム実行
- エラーコード
 - エラーコード一覧
- サポート

変更年月日	変更内容
2013-10-11	初版
2014-04-01	第2版 下記を追加・変更しました。 <ul style="list-style-type: none">ドキュメント全般 Windows Server 2012 向けの記述を追加しました。
2016-08-01	第3版 下記を追加・変更しました。 <ul style="list-style-type: none">「トラブルシューティング」→「原因と対処一覧」に注意事項を追加しました。
2018-12-01	第4版 下記を追加・変更しました。 <ul style="list-style-type: none">表記のゆれを訂正しました。
2019-04-01	第5版 下記を追加・変更しました。 <ul style="list-style-type: none">トラブルシューティングを本書から独立させました。
2020-04-01	第6版 下記を追加・変更しました。 <ul style="list-style-type: none">Windows 7 / Windows Server 2008 の記述を削除しました。「はじめに」のトラブルシューティングに関する記載を削除しました。
2020-08-01	第7版 下記を追加・変更しました。 <ul style="list-style-type: none">「サポート」の内容を見直しました。
2021-08-01	第8版 下記を追加・変更しました。 <ul style="list-style-type: none">「エラーコード一覧」へ注意を追加しました。
2021-12-01	第9版 下記を追加・変更しました。 <ul style="list-style-type: none">「PDFファイルの事前チェック」のサンプルプログラムをLinux対応版にしました。

目次

- 本書の目的
- 対象読者
- 本書の構成

本書の目的

本書では、IM-PDFCoordinator for Accel Platform を利用した基本的なプログラム開発や注意点等について説明します。

対象読者

本書は、開発をスムーズに開始するための手引書となっています。

したがって、実際に IM-PDFCoordinator for Accel Platform を利用したアプリケーションを開発するプログラマの方が対象です。

- 以下のいずれかを理解していることが必須です。
 - JavaEE開発モデル（Java）
 - スクリプト開発モデル（サーバサイドJavaScript）

また、本書は、以下に列挙する技術に関する知識を有することを前提として構成されています。

これらの技術に関して不明な点がある場合、本ドキュメントの内容を正しく理解することが困難になることがありますので、予めご了承ください。

なお、前提知識となる技術に関しては、一般の専門書籍等を参照してください。

- Javaプログラミング言語
- Java Servlet および JSP
- オペレーティングシステム
- ネットワーク

本書の構成

API リスト

利用できるAPIについて説明します。

プログラミング

プログラム開発の際の注意点や、プログラムの方法などを説明します。

チュートリアル

本製品のAPI を利用して実際にプログラムを作成する過程を学びます。

エラーコード

エラー発生時に返されるエラーコードを説明します。

サポート

製品サポートおよび技術情報の公開について説明します。

目次

- APIリストについて
- JavaEE開発モデル
- スクリプト開発モデル

APIリストについて

本製品には、IM-PDFCoordinator for Accel Platform 専用のAPI リストが付属します。

API リストは、document/apilist にあります。

IM-PDFCoordinator for Accel Platform には、JavaEE開発モデル 用のAPI が用意されています。

スクリプト開発モデル で開発をする場合は、スクリプト開発モデル のソースコード内でJavaのクラスを呼び出してください。

No.	フォルダ	説明
1	pdfprotection	セキュリティ機能API
2	pdfmakeup	セキュリティ機能「以外」のすべてのAPI

JavaEE開発モデル

IM-PDFCoordinator for Accel Platform は、JavaEE開発モデル で利用可能なJava-API（クラス）を用意しています。



スクリプト開発モデル

IM-PDFCoordinator for Accel Platform は、JavaEE開発モデル で利用可能なJava-API（クラス）を用意しています。

そのため、スクリプト開発モデル で IM-PDFCoordinator for Accel Platform を利用する場合は、スクリプト開発モデル のソースコード内でJavaのクラスを呼んでください。

スクリプト開発モデル 内でのJavaのクラスの呼び出し方法については、intra-mart Accel Platform 付属のマニュアルを参照ください。

目次

- 動作概念
- APIの種類と性質
- プログラム開発における注意点
 - PDFファイルへのアクセス
 - PDFファイルの事前チェック
- 体験版ライセンスにおける注意点

動作概念

通常の JavaEE開発モデル・スクリプト開発モデル プログラムは、ApplicationRuntime で実行されます。
IM-PDFCoordinator for Accel Platform で提供されるAPI も、ApplicationRuntime で動作します。
詳しくは、APIリストをご覧ください。

APIの種類と性質

IM-PDFCoordinator for Accel Platform は、JavaEE開発モデル で利用可能なJava-API（クラス）を用意しています。
そのため、スクリプト開発モデル で IM-PDFCoordinator for Accel Platform を利用する場合は、スクリプト開発モデル のソースコード内でJavaのクラスを呼んでください。
スクリプト開発モデル 内でのJavaのクラスの呼び出し方法については、intra-mart Accel Platform のマニュアルを参照ください。

プログラム開発における注意点

PDFファイルへのアクセス

IM-PDFCoordinator for Accel Platform が提供するAPIで加工・編集前後のファイルのパスを指定する際には、AppRuntimeからアクセス可能なパスを指定してください。

加工・編集するPDFファイルのサイズによっては、ネットワーク、APIのレスポンス、PDFファイルの編集が完全に終了するタイミングが大きく異なる場合があります。

特にサイズの大きいPDFファイルを加工・編集する場合は、十分な時間が経過した後に加工・編集したPDF ファイルにアクセスするようにしてください。

PDFファイルの事前チェック

外部から不特定のPDFファイルが投入されるシステムでは、サーバの安定運用の点からPDFファイルの事前チェックを推奨します。これは、PDFファイルに問題がないかチェックをすることで、サーバに害を与えるPDFファイルを事前にはじくことが目的です。

以下のサンプルでは、PDF結合処理を実行し正常終了するか確認をしています。

PDF事前チェックを exe で処理するのは、何か問題が発生した際に影響範囲をこのプロセス内に抑えるためです。

以下にサンプルを記載します。


```

1  import java.io.BufferedReader;
2  import java.io.IOException;
3  import java.io.InputStream;
4  import java.io.InputStreamReader;
5  import java.util.ArrayList;
6  import java.util.List;
7
8  /**
9   * PDF ファイルの結合処理を実行し、元ファイルに問題があるか判定します。
10  * サーバに悪影響を与えるファイルを事前にはじくことが目的です。
11  * 本ソースは、利用方法の説明のためのサンプルですのでサポート対象です。
12  * @version 1.0
13  */
14  public class pdfcheck {
15  /**
16  * ypdfcomb.exeを実行して元ファイルに問題があるかどうかチェックします。
17  * @param infile 元ファイル
18  * @param outfile 出力ファイル
19  * @param pwd パスワード
20  * @throws Exception エラー
21  */
22  public static void execute(String infile, String outfile, String pwd) throws Exception {
23  List<String> command = getCommandList(infile, outfile, pwd);
24
25  try {
26  ProcessBuilder pb = new ProcessBuilder(command);
27  // 標準エラーを標準出力にマージ
28  pb.redirectErrorStream(true);
29  Process p = pb.start();
30
31  InputStream is = null;
32  InputStreamReader sr = null;
33  BufferedReader br = null;
34  try {
35  is = p.getInputStream();
36  sr = new InputStreamReader(is);
37  br = new BufferedReader(sr);
38
39  String line;
40  StringBuilder message = new StringBuilder();
41  while ((line = br.readLine()) != null) {
42  message.append(line);
43  }
44
45  if (p.waitFor() != 0) {
46  throw new Exception(message.toString());
47  }
48  }
49  finally {
50  br.close();
51  sr.close();
52  is.close();
53  // 本来はチェックしたPDFファイルの削除が必要
54  }
55  }
56  catch (IOException e) {
57  throw new Exception(e.getMessage());
58  }
59  }
60
61  /**
62  * 実行用のコマンドを返す。
63  * @param infile 元ファイル
64  * @param outfile 出力ファイル
65  * @param pwd パスワード
66  * @return コマンド
67  */

```

```
68 private static List<String> getCommandList(String infile, String outfile,  
69 String pwd) {  
70 List<String> command = new ArrayList<String>();  
71 command.add("ypdfcomb");  
72 command.add("-se");  
73 command.add("y");  
74 command.add("-o");  
75 command.add(outfile);  
76 command.add("-temp");  
77 command.add(".");  
78 command.add("-i");  
79 command.add(infile);  
80  
81 return command;  
82 }  
83  
84 /**  
85  * サンプル実行  
86  */  
87 public static void main (String[] args) throws Exception {  
88  
89     pdfcheck.execute("C:/temp/in.pdf", "C:/temp/out.pdf", null);  
90 }  
91 }
```

体験版ライセンスにおける注意点

試用版ライセンスをご利用のお客様は、30～60 日間の試用期間が終了するとAPIが自動的に利用できない状態となります。

この状態でAPIを利用したプログラムを実行した場合に、実行時エラーとなります。

その場合は、正規の製品ライセンスを購入いただき、アンインストール後に再インストールしてください。

アンインストール・再インストールの方法は、インストールマニュアルをご確認ください。

目次

- 前提条件
- 用語解説
- 環境
 - サーバ環境
 - SOAPサーバ (Windows / セキュリティ機能用) 環境
 - 準備
- サンプルプログラムの場所 (すべての機能)
 - サンプルデータの用意
 - JSPプログラムの作成(セキュリティAPI)
- プログラム実行
 - 準備
 - プログラム実行
 - 確認

前提条件

このチュートリアルでは、JavaEE開発モデルにおけるプログラミングの方法について説明します。

このチュートリアルでは、セキュリティ機能 (利用するAPI は jp.co.iothe.pdfprotection パッケージ) のサンプルを作成します。

このチュートリアルで作成したプログラムは、C:/temp/sample.pdf ファイルをサンプルデータとして利用しますので、サンプルデータを作成してください。

このチュートリアルで作成したプログラムは、C:/temp ディレクトリにPDF ファイルを作成します。

用語解説

- Resin をインストールしたディレクトリを %RESIN_HOME% と略します。
- Apache HTTP Server をインストールしたディレクトリを %APACHE_HOME% と略します。
- Storage として使用するディレクトリを %PUBLIC_STORAGE_PATH% と略します。
- Webサーバ利用時の静的コンテンツを配置するディレクトリを %WEB_PATH% と略します。s
- PDFメイクアップ をインストールしたディレクトリを %PDFMAKEUP% と略します。

環境

チュートリアルを学ぶための環境です。

このドキュメント内では、ここで示す環境を前提として解説しています。

サーバ環境

intra-mart Accel Platform と IM-PDFCoordinator for Accel Platform が、正常にインストールされていることを前提とします。

SOAPサーバ (Windows / セキュリティ機能用) 環境

サーバには IM-PDFCoordinator for Accel Platform が正しくインストールされ、APIが正常に動作している状態であることが前提です。

サーバは Windows Server 2012 で動作しているものとして説明をします。

また、Java は Oracle JDK がインストールされているものとします。

準備

このドキュメントではC:\temp をプログラム作成の作業領域として説明しています。
 このフォルダが存在しない場合には、予め作成しておいてください。
 別なフォルダで作業をする場合には、その環境に合わせてドキュメントを読みすすめてください。
 プログラム作成には、テキストエディタが必要です。プログラム作成のできるテキストエディタをご用意下さい。

サンプルプログラムの場所（すべての機能）

%PDFMAKEUP%/sample/java

に機能毎のサンプルプログラムを用意しておりますので参照してください。

機能別フォルダ一覧表(%PDFMAKEUP%/sample/java 直下のフォルダ群)

No.	機能	サンプルフォルダ
1	保存場所制限	下記JSPプログラムを参照してください。
2	閲覧期限制御	下記JSPプログラムを参照してください。
3	パスワード付与	/samplesetProperty
4	パスワード解除	/samplesetProperty
5	PDFファイルの重ね合わせ	/samplemerge
6	PDFファイルにすかしを挿入	/sampletrans
7	用紙サイズの変更	/sampleedit
8	PDFファイルの結合	/samplecomb
9	PDFファイルの抽出・分割	/sampleextractpage、/samplediv
10	PDFファイルの回転	/samplediv
11	PDFファイルへの印鑑付与	/sampleiod、/sampletrans
12	PDFファイルへの文字・画像付与	/sampletrans
13	PDFファイルへのしおり・リンク付与	/sampleol
14	PDFファイルへのフォーム、注釈付与	/sampleform、/samplenote
15	PDFファイルへのJavaScript挿入	/sample1

サンプルデータの用意

%PDFMAKEUP%/sample/data

に機能に合わせてサンプルデータを用意しておりますので参照してください。

JSPプログラムの作成(セキュリティAPI)

このサンプルでは、JSPから IM-PDFCoordinator for Accel Platform の *セキュリティ機能* を呼び出します。

テキストエディタを起動して、以下のプログラムを記述します。
 この時、ファイル名の太文字・小文字は厳密な意味を持ちますので、注意してください。

作成した JSPファイルは、以下のフォルダに保存してください。

ファイル名	保存場所
protection.jsp	%RESIN_HOME%/webapps/{アプリケーション名}/


```

1  <%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
2
3  <%@ page import="java.util.Date" %>
4  <%@ page import="java.text.ParseException" %>
5  <%@ page import="java.text.SimpleDateFormat" %>
6
7  <%@ page import="jp.co.iothe.pdfprotection.PdfProtection" %>
8  <%@ page import="jp.co.iothe.pdfprotection.PdfProtectionException" %>
9  <%@ page import="jp.co.iothe.pdfprotection.PdfProtectionFactory" %>
10 <%
11 String src = "C:/temp/sample.pdf";
12 String pdf = "C:/temp/out.pdf";
13 String outpdf = "";
14 String message = "";
15
16 PdfProtection protection ;
17 int sts ;
18 String docinfname ;
19
20 // セキュリティ強化APIのインスタンスを生成します
21 // 設定ファイルに従ってリモートまたは直接使用のインスタンスを生成します
22 protection = PdfProtectionFactory.createPdfProtection();
23
24 // 文書情報を設定します
25 // このメソッドを実行しなかった場合、文書情報はすべて空になります
26 protection.setDocInfo(
27     "タイトル",
28     "サブタイトル",
29     "作成者",
30     "アプリケーション",
31     "キーワード"
32 );
33
34 // 標準セキュリティを設定します
35 // RC4-128ビットとAES128ビットのセキュリティはどちらか片方しか付与されません（最後に実行した方が有効）
36 // このメソッドを実行しなかった場合、標準セキュリティは付与されません
37 if( true ) {
38     System.out.println("RC4-128ビットのセキュリティ");
39     // RC4-128ビットのセキュリティを設定します
40     protection.setSecurity128("open", "security",
41         PdfProtection.SEC128PRINT_DISABLE,
42         PdfProtection.SEC128ACC_DISABLE,
43         PdfProtection.SEC128COPY_DISABLE,
44         PdfProtection.SEC128DOCCHANGE_DISABLE);
45 }
46 else {
47     System.out.println(" AES128ビットのセキュリティ");
48     // AES128ビットのセキュリティを設定します
49     protection.setSecurityAES128("open", "security",
50         PdfProtection.SEC128PRINT_DISABLE,
51         PdfProtection.SEC128ACC_DISABLE,
52         PdfProtection.SEC128COPY_DISABLE,
53         PdfProtection.SEC128DOCCHANGE_DISABLE);
54 }
55
56
57 // URLのセキュリティを設定します
58 // ワイルドカード「*」が使用できます
59 // このセキュリティを使用する場合、標準セキュリティで転載と文書変更を許可しないよう設定してください
60 // このメソッドを実行しなかった場合、URLのセキュリティは付与されません
61 String[] url = {"C:/temp/*", "http://localhost/*.pdf"};
62
63 for(int i = 0; i < url.length; i++){
64     System.out.println("url = " + url[i]);
65 }
66 protection.setSecurityUrl(url, "alert url");
67

```

```

68 // 有効期間のセキュリティを設定
69 // fromとtoのどちらか片方だけ設定することもできます
70 // このセキュリティを使用する場合、標準セキュリティで転載と文書変更を許可しないよう設定してください
71 // このメソッドを実行しなかった場合、有効期間のセキュリティは付与されません
72 //
73 // setSecurityDateでは、年月日までしか指定できません。
74 // 時間を指定しても、エラーになることはありませんが、無視されます。
75 // 例)2013/05/30~2013/05/30の場合、2013/05/30中はPDFを見ることができます。
76 String datePattern = "yyyy/MM/dd";
77 SimpleDateFormat sdf = new SimpleDateFormat(datePattern);
78 Date fromDate = null;
79 Date toDate = null;
80 try {
81     fromDate = sdf.parse("2013/07/01");
82     toDate = sdf.parse("2013/08/31");
83     System.out.println(fromDate.toString());
84     System.out.println(toDate.toString());
85 }
86 catch (ParseException e) {
87     e.printStackTrace();
88 }
89
90 protection.setSecurityDate(null, toDate, "alert date");
91
92 // Webに最適化するかどうかを設定します
93 protection.setFastWebView(true);
94
95 // 上記で設定した情報を元にセキュリティを強化して新しいPDFを出力します
96 int result;
97 if (true) {
98     // 編集元PDFにセキュリティパスワードが設定されていない場合
99     result = protection.outputPdf(src, pdf);
100 }
101 else {
102     // 編集元PDFにセキュリティパスワードが設定されている場合
103     result = protection.outputPdf(src, "password", pdf);
104 }
105
106 // outputPdfで発生した例外を取得します
107 if (result < 0) {
108     PdfProtectionException exception = protection.getException();
109     message = exception.getMessage();
110 }
111 %>
112 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 //EN">
113 <HTML>
114 <HEAD>
115 <TITLE>My sample</TITLE>
116 </HEAD>
117 <BODY bgcolor="WhiteSmoke">
118 <CENTER>
119 <%
120 if ("".equals(message)){
121     out.println("<H1>SUCCESS!!</H1>");
122     out.println("<H2>" + pdf + "</H2>");
123 }
124 else{
125     out.println("<H1>FAILURE!!</H1>");
126     out.println("<H2>" + message + "</H2>");
127 }
128 %>
129 </H2>
130 </CENTER>
131 </BODY>
132 </HTML>

```



コラム

標準セキュリティはRC4-128ビットとAES128ビットのセキュリティはどちらか片方しか付与されません。（最後に実行した方が有効になります。）



コラム

URLのセキュリティはワイルドカード「*」が使用できます。このセキュリティを使用する場合、標準セキュリティで転載と文書変更を許可しないよう設定してください。



コラム

有効期限のセキュリティはfromとtoのどちらか片方だけ設定することもできます。このセキュリティを使用する場合、標準セキュリティで転載と文書変更を許可しないよう設定してください。setSecurityDateでは、年月日までしか指定できません。

プログラム実行

準備

実行させるための準備の手順を説明します。

サンプルファイルの用意

1. intra-mart サーバが稼働中の場合は停止状態にします。
2. 転送したクラスファイルをAppRuntime の動作するService-Platform をインストールしたディレクトリ内の doc/imart/WEB-INF/classes/myapp ディレクトリに保存してください。
3. intra-mart サーバを起動します。

メニュー設定

1. テナント管理者でログインし、以下のメニューを設定します。
2. [テナント管理]-[メニュー]画面を開きます。
3. フォルダを作成します。

4. URLに、protection.jsp を設定し、メニューを追加します。

メニューアイテムの編集

メニューアイテムID * 5i7j7cu000dion1

メニューアイテム名 *

日本語 *	protection
英語	protection
中国語	protection

URL * protection.jsp 権限設定

呼び出し方法 GET

引数

+ 行追加 - 選択行削除

キー	値

アイコン画像

ファイルパス コンテキストパス配下のURLを入力してください。

CSS Sprites imui://csssprites/ クラス名を入力してください。

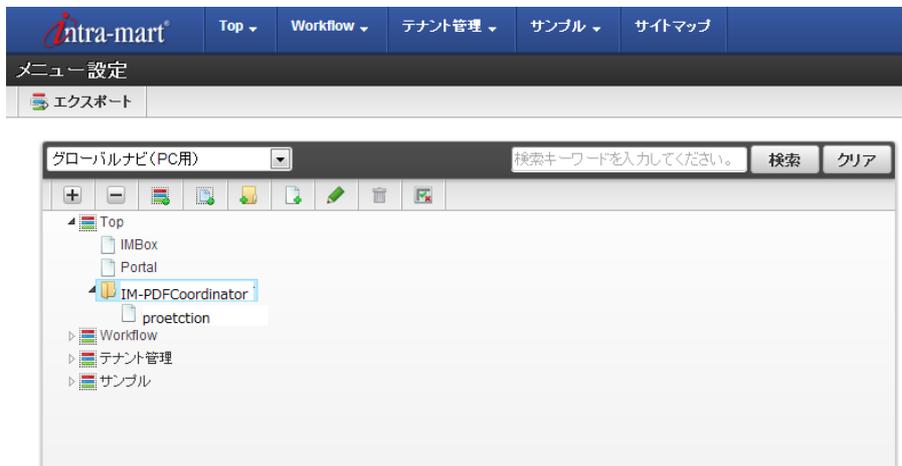
IFRAME表示

ポップアップ表示

説明

更新

5. メニュー設定は完了です。



プログラム実行

メニューで『protection』を選択してください。作成したJSPファイルが実行されます。

JSPの実行エラー（コンパイルエラー）になってしまった場合には、エラーメッセージの内容に従いJSPプログラムを修正してください。

JSPプログラムが正しく動作しているにも関わらず実行時エラーになってしまう場合は、エラーの内容にしたがって環境を正しく構築してください（環境を変更した場合は、サーバの再起動が必要になる場合があります）。

確認

プログラムが正しく実行されると IM-PDFCoordinator for Accel Platform の C:/temp ディレクトリに out.pdf というファイル名のPDFファイルが作成されます。

このファイルがPDFビューア（Adobe AcrobatReader など）で正しく表示できればすべての処理が正しく行われたこととなります。

目次

- エラーコード一覧

エラーコード一覧

ステータスコード	エラー内容
-1	編集元PDFのパスが指定されていません。
-2	編集元PDFが見つかりません。
-3	編集元PDFに問題があるため処理できません。
-4	出力先PDFのパスが指定されていません。
-5	一時フォルダの作成に失敗しました。
-6	一時ファイルの作成に失敗しました。
-7	ypdfcombコマンドの実行に失敗しました。
-8	致命的エラーが発生した為、除外リストに追加しました。PDFメイクアップでエラーが発生しました。
-9	Javascriptファイルの編集に失敗しました。
-10	ファイルの作成に失敗しました。
-11	リソースのコピーに失敗しました。
-12	利用可能なサーバが見つかりません。
-13	PDFセキュリティ強化サービスでエラーが発生しました。
-100	システムエラーが発生しました。



コラム

PDFエンジン部分のエラーについては、スタート→YSS PDF Makeup→ドキュメント→エラー一覧 から確認ください。



注意

「Password error」と表示された場合でも、パスワードに起因するエラーではない場合があります。

上記のエラーが発生する主なPDFファイルの形式やケースは、次の通りです。

- パスワードが付与され、暗号化されているPDFファイル
- 電子署名やタイムスタンプ等が付与されているPDFファイル
- Adobe Acrobat の拡張機能等が使用されているPDFファイル
- 内部構造が一部破損しているPDFファイル
- PDFの規格に準拠していないPDFファイル

弊社では、Web にて弊社製品に対するサポートおよび技術情報の公開を行っております。

当製品に関して不明な点などがございましたら、情報検索または弊社サポート窓口までご相談ください。