

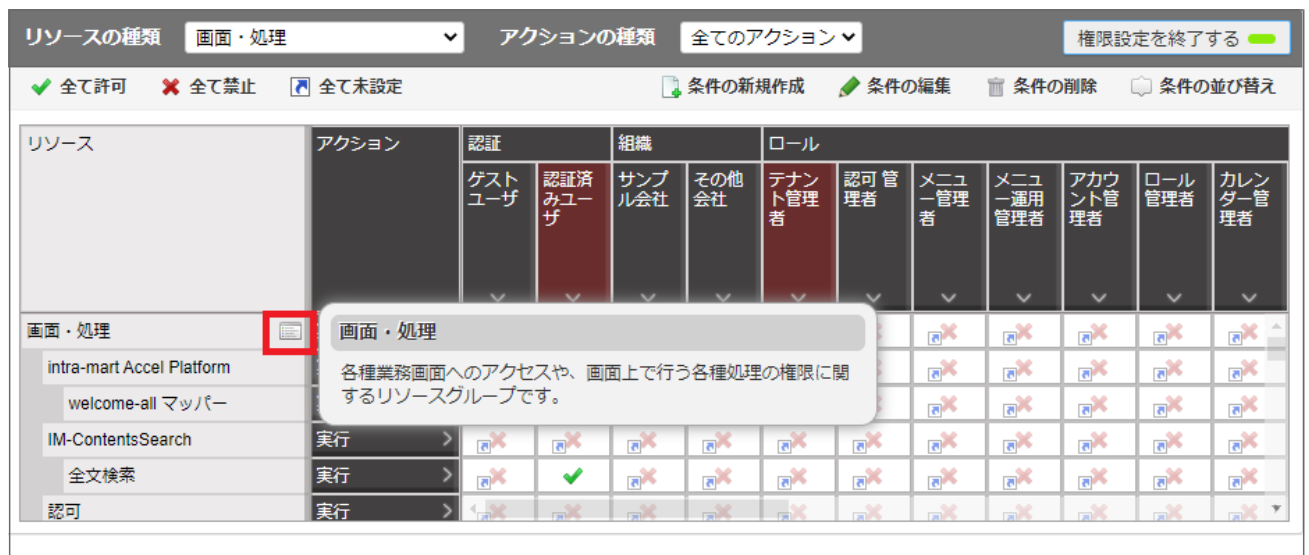
1. テナント管理者でログインし、次のメニューを設定します。

2. [テナント管理]-[認可]画面を開きます。

3. [権限設定を開始する]ボタンを押下します。



4. [リソース]を選択し、[リソースの詳細を開く]押下します。



5. [配下にリソースを新規作成]を押下します。

リソースの詳細

配下にリソースを新規作成 リソースの編集 リソースの削除 リソースの閉塞

リソースグループID	http-services												
リソースグループ名	画面・処理												
リソースURI													
説明	各種業務画面へのアクセスや、画面上で行う各種処理の権限に関するリソースグループです。												
リソースの階層	<table><thead><tr><th>リソースグループ名</th><th>開く</th></tr></thead><tbody><tr><td>画面・処理</td><td></td></tr><tr><td>intra-mart Accel Platform</td><td></td></tr><tr><td>IM-ContentsSearch</td><td></td></tr><tr><td>認可</td><td></td></tr><tr><td>カレンダー</td><td></td></tr></tbody></table>	リソースグループ名	開く	画面・処理		intra-mart Accel Platform		IM-ContentsSearch		認可		カレンダー	
リソースグループ名	開く												
画面・処理													
intra-mart Accel Platform													
IM-ContentsSearch													
認可													
カレンダー													

6. リソースグループを作成します。

リソースの設定

リソースグループID *	8gokmt7bq3mvra0		
リソースグループ名 *	日本語 *		
	英語		
	中国語 (中国)		
リソースURI			
	リソースURIを省略するとリソースグループとして、入力するとリソースとして登録します。		
説明	日本語		
	英語		
	中国語 (中国)		

OK

7. リソースグループ名に、次の値を設定します。

機能	リソースグループ名
ページ機能 (Page)	Script_PDF結合サンプル
マージ機能 (Merge)	Script_PDFマージサンプル
エディット機能 (Edit)	Script_PDF追記サンプル

リソースの設定

リソースグループID *

8h0u8b551ycz0sg

リソースグループ名 *

日本語 *

Script_PDF結合サンプル

英語

Script_PDF結合サンプル

中国語 (中華人民共和国)

Script_PDF結合サンプル

リソースURI

リソースURIを省略するとリソースグループとして、入力するとリソースとして登録します。

説明

日本語

英語

中国語 (中華人民共和国)

OK

8. リソースURIに、次の値を設定します。

機能	リソースURI
ページ機能 (Page)	service://pdfc/script/combine
マージ機能 (Merge)	service://pdfc/script/merge
エディット機能 (Edit)	service://pdfc/script/edit

リソースの設定

リソースグループID *

8h0u8b551ycz0sg

リソースグループ名 *

日本語 *

Script_PDF結合サンプル

英語

Script_PDF結合サンプル

中国語 (中華人民共和國)

Script_PDF結合サンプル

リソースURI

service://pdfc/script/combine

?

リソースURIを省略するとリソースグループとして、入力するとリソースとして登録します。

説明

日本語

英語

中国語 (中華人民共和國)

OK

9. 作成したリソースグループで「認証済みユーザ」に「全て許可」を付与します。

リソース	アクション	認証		組織	
		ゲストユーザ	認証済みユーザ	サンプル会社	その他会社
画面・処理	実行	▼	▼	▼	▼
intra-mart Accel Platform	実行	✖	✖	✖	✖
welcome-all マップ	実行	✔	✔	✖	✖
Script_PDF結合サンプル	実行	✖	✔	✖	✖
IM-ContentsSearch	実行	✖	✖	✖	✖
全文検索	実行	✖	✔	✖	✖

メニュー設定

- テナント管理者でログインし、次のメニューを設定します。
- [テナント管理]-[メニュー]画面を開きます。
- メニューフォルダを作成します。
同一のメニューフォルダを既に設定している場合、当該手順は不要です。

メニューフォルダの新規作成

メニューフォルダID *

8gokmzl0d3n94a0

メニューフォルダ名 *

日本語 *

IM-PDFCoordinator

英語

IM-PDFCoordinator

中国語 (中国)

IM-PDFCoordinator

アイコン画像

標準

☒ 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。

16px



32px



48px



新規作成

4. 作成したフォルダの下にメニューアイテムを新規作成し、メニューアイテム名、および、URLに次の値を設定します。

機能	メニューアイテム名	URL
ページ機能 (Page)	Script_combine	pdfc/script/combine
マージ機能 (Merge)	Script_merge	pdfc/script/merge
エディット機能 (Edit)	Script_edit	pdfc/script/edit

メニューアイテムの新規作成

メニューアイテムID

8h0u8kzv9ydd3sg

メニューアイテム名

日本語

Script_combine

英語

Script_combine

中国語 (中華人民共和国)

Script_combine

URL

pdfc/script/combine

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

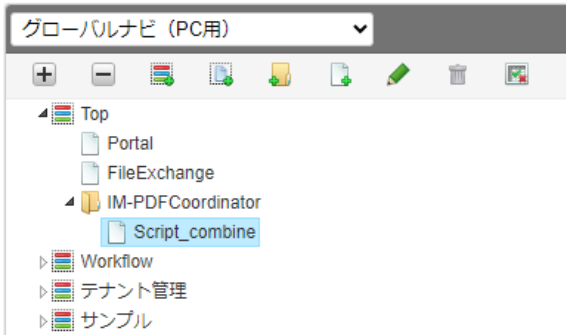
標準

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。

新規作成

5. メニュー設定は完了です。



プログラムの実行と確認

メニューで次のアイテムを選択することにより、作成した画面が表示されます。

機能	メニューアイテム名
ページ機能 (Page)	Script_combine
マージ機能 (Merge)	Script_merge
エディット機能 (Edit)	Script_edit

画面上で処理対象ファイルをアップロードすることで、編集/加工処理のプログラムが実行され、処理されたPDFファイルがダウンロードされます。

PDFビューア (Adobe Acrobat Reader など) で処理後のファイルが正しく表示されることを確認し、このチュートリアルは完了です。

目次

■ エラーコード一覧

■ PDF Makeup

■ PDF Protection

エラーコード一覧

PDF Makeup

PDF Makeupのエラーコードについては、連携エンジン PDFメイクアップ をインストールしたサーバの「スタート」→「YSS PDF Makeup」→「ドキュメント」→「エラー一覧」から確認してください。

!

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ケースがあります。

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

PDF Protection

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。

Webにて当製品に対するサポート、および、技術情報を公開しています。

当製品に関して不明な点などがある場合、情報検索、または、「[intra-mart サポートサイト](#)」に問い合わせしてください。

IM-PDFCoordinator for Accel Platform を使って IM-LogicDesigner でPDFファイルを結合する方法

本項では、IM-LogicDesigner の JavaScript定義 で、IM-PDFCoordinator for Accel Platform のAPIを使用したPDF結合について紹介しています。



注意

本サンプルは、IM-PDFCoordinator for Accel Platform 2024 Spring 以降のバージョンが必要です。

本サンプルを実施するにあたり、次のzipファイルをダウンロードし、解凍してください。

< [sample_combine_ld.zip](#) >

解凍したファイルの構成は、次の通りです。

フォルダ名／ファイル名	説明
data / comb1.pdf comb2.pdf	PDF結合サンプル用のPDFファイル
import / im_logicdesigner-data.zip	PDF結合用ユーザ定義、フロー定義をまとめたzipファイル

< import/im_logicdesigner-data.zip >を、IM-LogicDesigner のインポート画面からインポートしてください。

本サンプルのユーザ定義（JavaScript定義）では、IM-PDFCoordinator for Accel Platform の スクリプト開発モデル 用APIを使用しています。

サンプルの実行手順、ユーザ定義の詳細、および、サンプル内で使用しているAPIについては、次を参照してください。

サンプル実行手順

- [結合対象ファイルを設置する](#)
- [サンプルのフロー定義のデバッグ画面を開く](#)
- [入力値を設定し、デバッグを実行する](#)
- [実行結果を確認する](#)

[結合対象ファイルを設置する](#)

1. < %PUBLIC_STORAGE_PATH% /pdfc_ld >ディレクトリを作成します。
2. サンプル用のPDFファイルが入っている< data >フォルダを、< %PUBLIC_STORAGE_PATH% /pdfc_ld >配下に設置します。

[サンプルのフロー定義のデバッグ画面を開く](#)

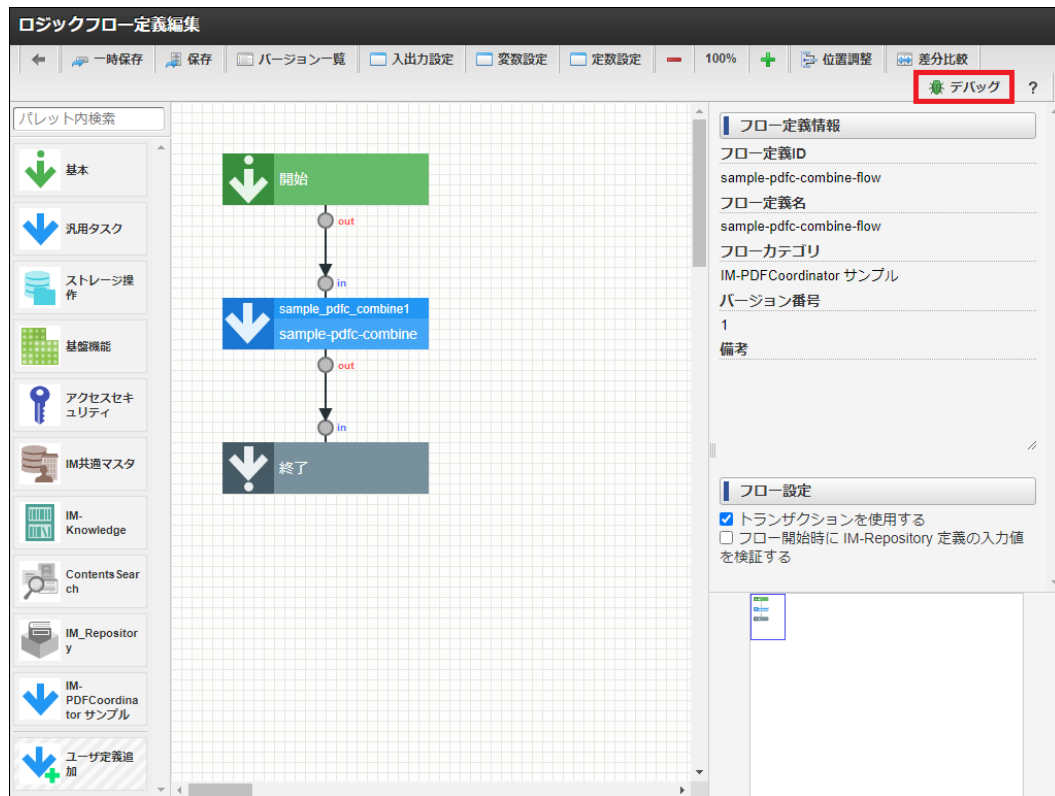
1. 「サイトマップ」 - 「LogicDesigner」 - 「フロー定義一覧」をクリックします。



2. 「IM-PDFCoordinator サンプル」 - 「sample-pdfc-combine-flow」を選択し、「編集」ボタンをクリックします。

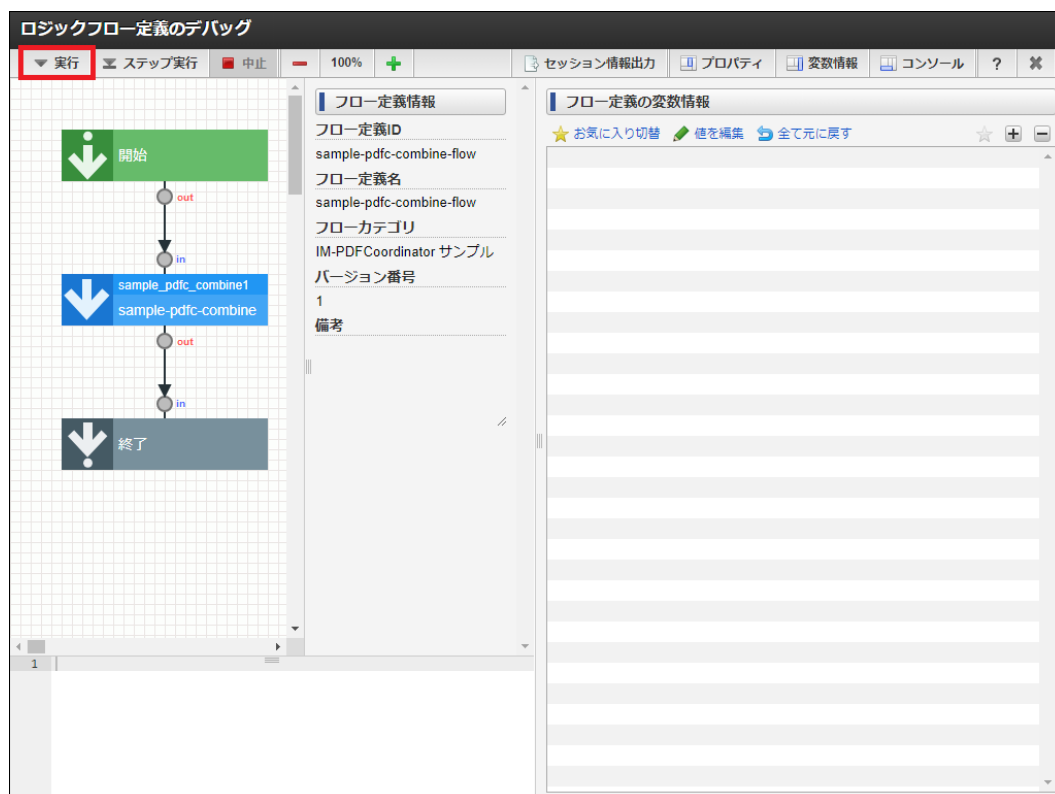


3. 「デバッグ」ボタンをクリックします。



入力値を設定し、デバッグを実行する

1. 「実行」ボタンをクリックします。



2. 「combFile1Path」、および、「combFile2Path」に結合対象ファイル、「outputFilePath」に出力ファイルのパブリックストレージパスを設定し、「実行」ボタンをクリックします。

設定例は次の通りです。

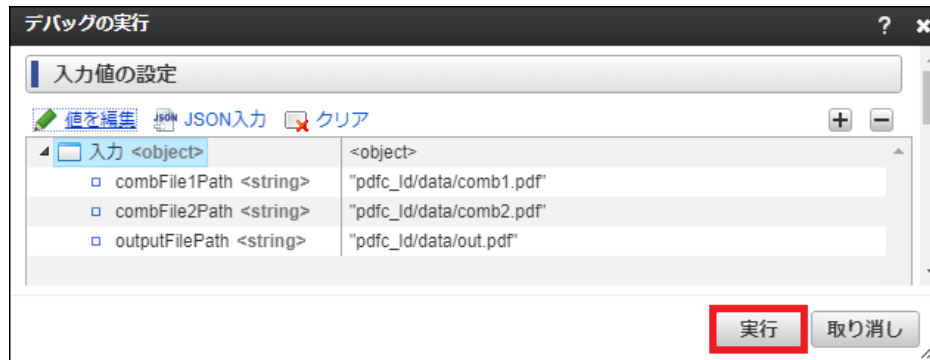
< 値を編集 >

変数	値
combFile1Path	pdfc_id/data/comb1.pdf
combFile2Path	pdfc_id/data/comb2.pdf

変数	値
outputFilePath	pdfc_ld/data/out.pdf

< JSON入力 >

```
{
  "combFile1Path": "pdfc_ld/data/comb1.pdf",
  "combFile2Path": "pdfc_ld/data/comb2.pdf",
  "outputFilePath": "pdfc_ld/data/out.pdf"
}
```



i コラム
ユーザ定義の入出力値については「[ユーザ定義タスク](#)」を参照してください。

i コラム
本フロー定義では、次の順番でファイルが結合されます。

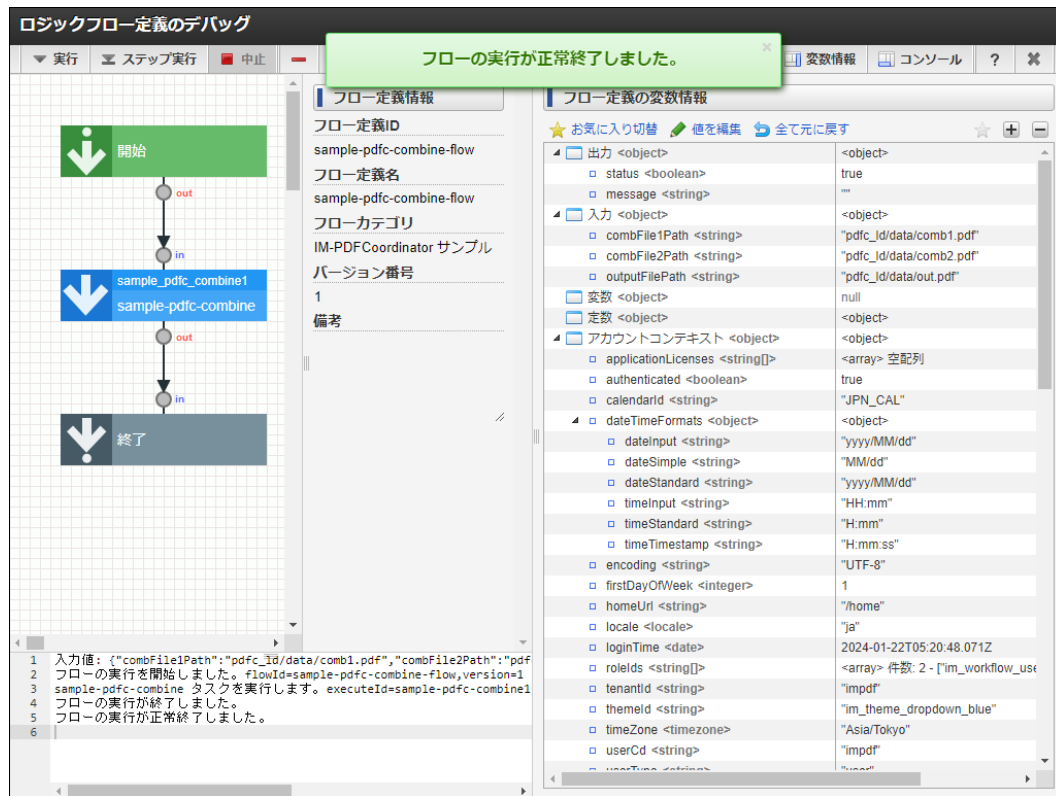
- combFile1Path
- combFile2Path

3. 「決定」ボタンをクリックします。



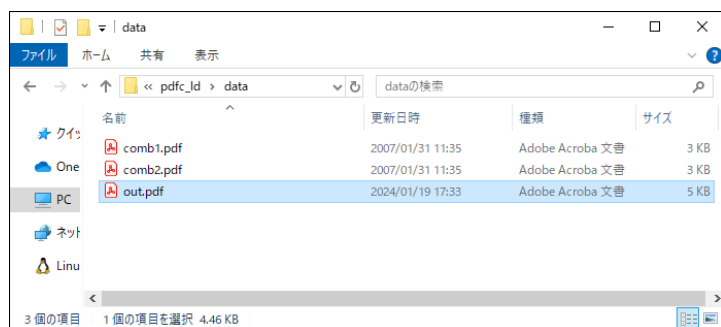
4. デバッグが開始されます。

正常にデバッグが終了した場合、その旨のメッセージが表示され、変数情報ペイン、および、コンソールペインが更新されます

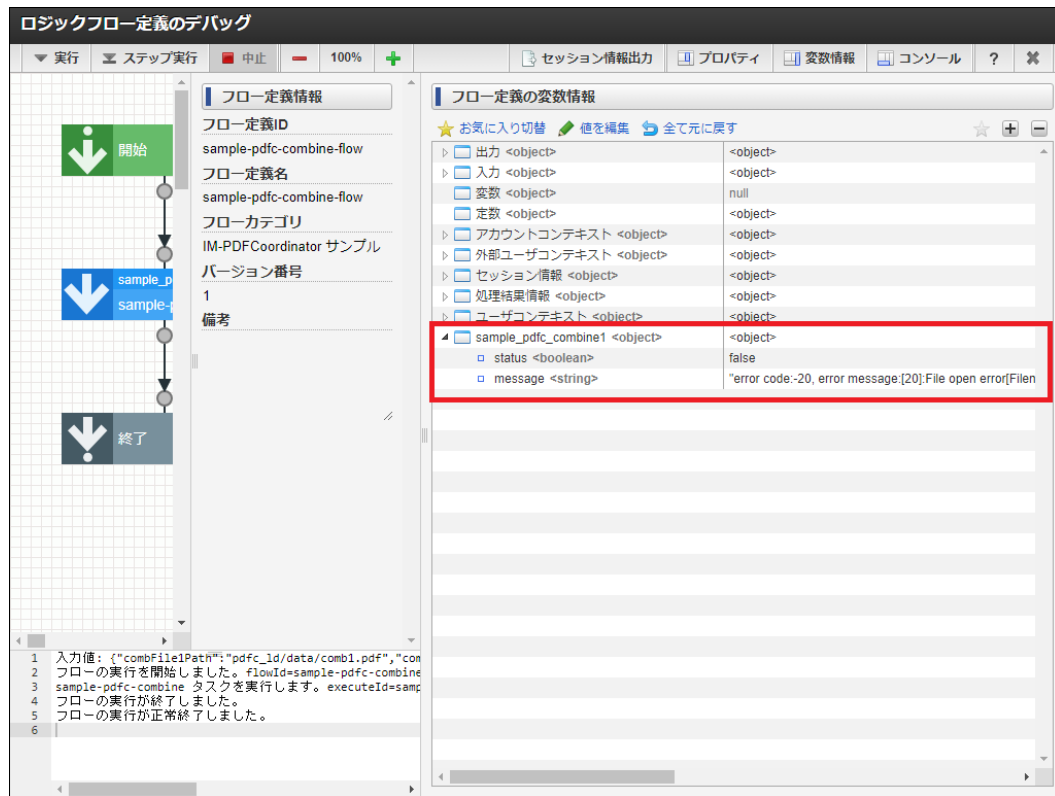


実行結果を確認する

1. 「outputFilePath」に指定した出力先に、PDFファイルが出力されていることを確認します。



PDFの結合処理に失敗した場合は、デバッグ実行時のユーザ定義の返却値「status」、および、「message」を確認してください。



以上で、全ての手順は終了です。

ユーザ定義タスク

PDF変換

編集	ユーザ定義ID	ユーザ定義名	種別	ユーザカテゴリ	呼出元
	sample-pdfc-combine	sample-pdfc-combine	javascript	IM-PDFCoordinator サンプル	

入力値

```
sample_pdfc_combine <object>
├ combFile1Path <string>
├ combFile2Path <string>
└ outputPath <string>
```

項目名	必須/任意	型	配列/リスト	説明
combFile1Path combFile2Path	必須	string	なし	PDF結合対象ファイルのパブリックストレージパス
outputFilePath	必須	string	なし	PDF結合出力ファイルのパブリックストレージパス

出力値

```
sample_pdfc_combine <object>
├ status <boolean>
└ message <string>
```

項目名	型	配列/リスト	説明
status	boolean	なし	true : PDF結合成功時 false : PDF結合失敗時
message	string	なし	PDF結合成功時 : 空文字 PDF結合失敗時 : エラーメッセージ

サンプル内で使用しているAPIについては「[API](#)」を参照してください。

 コラム

文書情報を設定する場合は、スクリプトの81、82行目のコメントを外してください。

 コラム

セキュリティ情報を設定する場合は、設定したいセキュリティの暗号化レベルに合わせて、スクリプトの次のコメントを外してください。

- 40bit RC4 : 93、94行目
- 128bit RC4 : 116、117行目
- 128bit AES : 139、140行目

```

1  /**
2   * run.
3   *
4   * @param input {Object} - task input data.
5   * @return {Object} task result.
6   */
7  function run(input) {
8
9      const tempFiles = new IotheCommonTempFiles();
10
11      try {
12          if (!input.combFile1Path) {
13              throw new Error("処理対象PDFファイルパスにnull、または、空文字が指定されています。");
14          }
15
16          if (!input.combFile2Path) {
17              throw new Error("処理対象PDFファイルパスにnull、または、空文字が指定されています。");
18          }
19
20          if (!input.outputFilePath) {
21              throw new Error("出力対象PDFファイルパスにnull、または、空文字が指定されています。");
22          }
23
24          if ((input.combFile1Path.indexOf('.') !== -1) && (input.combFile2Path.indexOf('.') !== -1)) {
25              const srcFileExt1 = "." + input.combFile1Path.split('.').pop();
26              const srcFileExt2 = "." + input.combFile2Path.split('.').pop();
27              const tempCombFile1 = tempFiles.copyFrom(input.combFile1Path, srcFileExt1);
28              const tempCombFile2 = tempFiles.copyFrom(input.combFile2Path, srcFileExt2);
29              const tempOutFile = tempFiles.create();
30
31              // IM-PDFCoordinatorを実行し、PDFファイルを結合します。
32              execPdfcoordinatorCombine(tempCombFile1.path(), tempCombFile2.path(), tempOutFile.path());
33
34              tempFiles.copyTo(tempOutFile, input.outputFilePath);
35
36              return {
37                  status: true,
38                  message: ""
39              };
40          } else {
41              throw new Error("処理対象PDFファイルの拡張子がありません。");
42          }
43      } catch (error) {
44          return {
45              status: false,
46              message: error.message
47          };
48      } finally {
49          tempFiles.close();
50      }
51  }
52
53  /**
54   * IM-PDFCoordinatorを実行し、PDFファイルを結合します。
55   * @param {String} combFile1Path 被結合対象のファイルパス
56   * @param {String} combFile2Path 結合対象のファイルパス
57   * @param {String} outFilePath 結合後の出力先PDFファイルパス
58   */
59  function execPdfcoordinatorCombine(combFile1Path, combFile2Path, outFilePath)
60  {
61      // PDFをファイル単位で結合するクラスのインスタンスを生成します。
62      // @return {Object} PDFをファイル単位で結合するクラスのインスタンス
63      const comb = new pdfcombine();
64      checkerror(0, comb);
65
66      // エンコード文字列を指定します。
67      comb.m_encode = "MS932";
68
69      // 内部メンバの初期化等を行います。
70      // @returns {Number} 正常時は0、エラー時は-1を返します。
71      let sts = comb.init();
72      checkerror(sts, comb);
73
74      // 出力PDFの文書情報を設定します。

```

```

75 // setdocinfo(title, subTitle, creator, app, keyword);
76 // @param {String} title タイトルに設定する文字列を指定します。
77 // @param {String} subtitle サブタイトルに設定する文字列を指定します。
78 // @param {String} creator 作成者に設定する文字列を指定します。
79 // @param {String} app 作成アプリケーションに設定する文字列を指定します。
80 // @param {String} keyword キーワードに設定する文字列を指定します。
81 // sts = comb.setdocinfo("文書タイトル", "文書サブタイトル", "作成者", "作成アプリケーション名", "キーワード");
82 // checkerror(sts, comb);
83
84 // 出力PDFのRC4 40ビットセキュリティを設定します。
85 // setsecurity(fromtop, showpasswd, securitypasswd, noprint, noedit, nocopy, noaddnote);
86 // @param {boolean} fromtop 連結元の先頭のPDFを引継ぎます。
87 // @param {String} showpasswd 参照用のパスワードを指定します。
88 // @param {String} securitypasswd セキュリティ設定用のパスワードを指定します。
89 // @param {boolean} noprint 印刷(true: 不可, false: 可能)
90 // @param {boolean} noedit 編集(true: 不可, false: 可能)
91 // @param {boolean} nocopy 転載(true: 不可, false: 可能)
92 // @param {boolean} noaddnote 注釈追加(true: 不可, false: 可能)
93 // sts = comb.setsecurity(false, "open", "security", false, true, false, true);
94 // checkerror(sts, comb);
95
96 // 出力PDFのRC4 128ビットセキュリティを設定します。
97 // setsecurity128(showpasswd, securitypasswd, print, access, copy, change);
98 // @param {String} showpasswd 参照用のパスワードを指定します。
99 // @param {String} securitypasswd セキュリティ設定用のパスワードを指定します。
100 // @param {String} print 128bit security(印刷)を指定します。
101 // "PRINT_DISABLE": 許可しない
102 // "PRINT_DEGRADED": 低解像度で許可する
103 // "PRINT_ENABLE": 許可する
104 // @param {String} access 128bit security(アクセス)を指定します。
105 // "ACC_DISABLE": 許可しない
106 // "ACC_ENABLE": 許可する
107 // @param {String} copy 128bit security(転載)を指定します。
108 // "COPY_DISABLE": 許可しない
109 // "COPY_ENABLE": 許可する
110 // @param {String} change 128bit security(文書変更)を指定します。
111 // "DOCCHANGE_DISABLE": 許可しない
112 // "DOCCHANGE_ASSEMBLE": アセンブリを許可する
113 // "DOCCHANGE_FORMFILL": フォーム入力を許可する
114 // "DOCCHANGE_ADDNOTE": フォーム入力と注釈追加を許可する
115 // "DOCCHANGE_ENABLE": 許可する
116 // sts = comb.setsecurity128("open", "security", "PRINT_DEGRADED", "ACC_ENABLE", "COPY_DISABLE",
117 "DOCCHANGE_ADDNOTE");
118 // checkerror(sts, comb);
119
120 // 出力PDFのAES 128ビットセキュリティを設定します。
121 // setsecurityaes128(showpasswd, securitypasswd, print, access, copy, change);
122 // @param {String} showpasswd 参照用のパスワードを指定します。
123 // @param {String} securitypasswd セキュリティ設定用のパスワードを指定します。
124 // @param {String} print 128bit security(印刷)を指定します。
125 // "PRINT_DISABLE": 許可しない
126 // "PRINT_DEGRADED": 低解像度で許可する
127 // "PRINT_ENABLE": 許可する
128 // @param {String} access 128bit security(アクセス)を指定します。
129 // "ACC_DISABLE": 許可しない
130 // "ACC_ENABLE": 許可する
131 // @param {String} copy 128bit security(転載)を指定します。
132 // "COPY_DISABLE": 許可しない
133 // "COPY_ENABLE": 許可する
134 // @param {String} change 128bit security(文書変更)を指定します。
135 // "DOCCHANGE_DISABLE": 許可しない
136 // "DOCCHANGE_ASSEMBLE": アセンブリを許可する
137 // "DOCCHANGE_FORMFILL": フォーム入力を許可する
138 // "DOCCHANGE_ADDNOTE": フォーム入力と注釈追加を許可する
139 // "DOCCHANGE_ENABLE": 許可する
140 // sts = comb.setsecurityaes128("open", "security", "PRINT_DEGRADED", "ACC_ENABLE", "COPY_DISABLE",
141 "DOCCHANGE_ADDNOTE");
142 // checkerror(sts, comb);
143
144 // PDF出力後のWebに最適化の処理の有無を設定します。
145 // 特にこのメソッドを呼び出さない場合はデフォルトで最適化されます。
146 // setfastwebview(bfastwebview);
147 // @param {boolean} bfastwebview true: 最適化する, false: 最適化しない
148 sts = comb.setfastwebview(true);
149 checkerror(sts, comb);

```

```

150
151 // 出力PDFをオープンし、連結の準備をします。
152 // open(outpdf);
153 // @param {String} outpdf 出力先PDFのファイル名を指定します。
154 sts = comb.open(outFilePath);
155 checkerror(sts, comb);
156
157 // 指定ファイルのPDF連結の準備をします。
158 // combine(pdf);
159 // @param {String} pdf 連結するPDFのファイル名を指定します。
160 sts = comb.combine(combFile1Path);
161 checkerror(sts, comb);
162 sts = comb.combine(combFile2Path);
163 checkerror(sts, comb);
164
165 // 出力PDFを連結及びクローズし、連結を終了します。
166 sts = comb.close();
167 checkerror(sts, comb);
168
169 // 内部のハンドルを開放します。
170 comb.release();
171 }
172
173 /**
174 * メソッドの戻り値からエラーを判定し、エラーであれば例外を投げます。
175 * @param {Number} sts メソッドの戻り値
176 * @param {Object} obj メソッドを実行したインスタンス
177 */
178 function checkerror(sts, obj) {
179     if (obj === null) {
180         throw new Error("did not create the object");
181     }
182     if (sts < 0) {
183         throw new Error("error code:" + obj.geterrorno() + ", error message:" + obj.geterror());
184     }
185 }

```

API

サンプル内で使用しているAPIについて示します。

pdfcombine

pdfcombineクラスの詳細については、「[IM-PDFCoordinator for Accel Platform - PDF Makeup API ドキュメント](#)」- スクリプト開発モデル「[pdfcombine](#)」を参照してください。

lotheCommonTempFiles

lotheCommonTempFilesクラスの詳細については、「[lotheCommonTempFiles](#)」を参照してください。

lotheCommonTempFiles

一時ファイルを操作するクラスです。

一時ファイルは、「[intra-mart Accel Platform 設定ファイルリファレンス](#)」-「[一時ファイルディレクトリ](#)」で設定したディレクトリ配下に作成されます。



コラム

本クラスの使用例については、「[スクリプト](#)」を参照してください。



注意

本クラスは、IM-PDFCoordinator for Accel Platform 2024 Spring 以降のバージョンで使用可能です。

2023 Autumn 以前のバージョンの場合は、2024 Spring 以降のバージョンにアップデートしてください。

Constructor

インスタンスオブジェクトを作成します。

Returns

生成されたインスタンスオブジェクト

Methods

create(ext)

空のファイルを作成します。

Parameters

Name	Type	Description
ext	String	ファイル名を生成するために使用される接尾辞文字列 null、または、未指定時は".tmp" が使用されます。

Returns

新規作成された空のファイルを示すFileオブジェクト

Exception

以下のエラーメッセージを持つErrorオブジェクト

- 指定された引数extの値が不正な場合：「接尾辞文字列の値が不正です。」

copyFrom(srcFilePath, ext)

指定したファイルをコピーし、コピー先のファイルを返します。

Parameters

Name	Type	Description
srcFilePath	String	コピー元の対象ファイルのパブリックストレージパス
ext	String	コピー先のファイル名を生成するために使用される接尾辞文字列 null、または、未指定時は".tmp" が使用されます。

Returns

コピー先ファイルを示すFileオブジェクト

Exception

以下のエラーメッセージを持つErrorオブジェクト

- 指定された引数srcFilePathの値が不正な場合：「コピー元の対象ファイルパスの値が不正です。」
- 指定された引数extの値が不正な場合：「接尾辞文字列の値が不正です。」
- ファイルのコピーに失敗した場合：「ファイルのコピーに失敗しました。」

copyTo(tempFile, destFilePath)

指定したファイルを、指定先へコピーします。

Parameters

Name	Type	Description
tempFile	File	コピー元の対象ファイル
destFilePath	String	コピー先の対象ファイルのパブリックストレージパス

Exception

以下のエラーメッセージを持つErrorオブジェクト

- 指定された引数tempFileの値が不正な場合：「コピー元の対象ファイルの値が不正です。」
- 指定された引数destFilePathの値が不正な場合：「コピー先の対象ファイルパスの値が不正です。」
- ファイルのコピーに失敗した場合：「ファイルのコピーに失敗しました。」

close()

一時ファイルを削除し、一時ファイルに関する操作を終了します。



注意

当該メソッドを実行しない場合、一時ファイルディレクトリに一時ファイルが蓄積されていきます。

一時ファイルを残す必要が無い場合は、finallyブロック等を使い、必ず当該メソッドを実行して一時ファイルを削除してください。

Exception

以下のエラーメッセージを持つErrorオブジェクト

- 一時ファイルの削除に失敗した場合：「一時ファイルの削除に失敗しました。」