



# FormPat API 開発ガイド



2022/08/24

**Digital Assist**

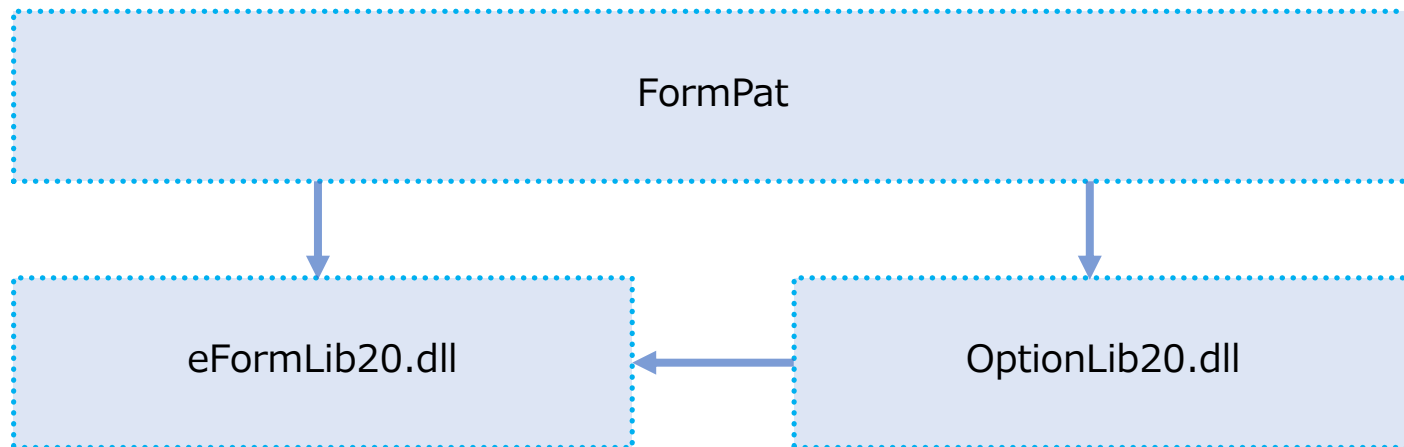
Copyright© 2022 Digital Assist Corporation. All Rights Reserved.

# FormPat API の概要

FormPat API は、FormPat 本体のプログラム改修なしに、FormPat の各種イベントで独自処理の開発を可能にします。

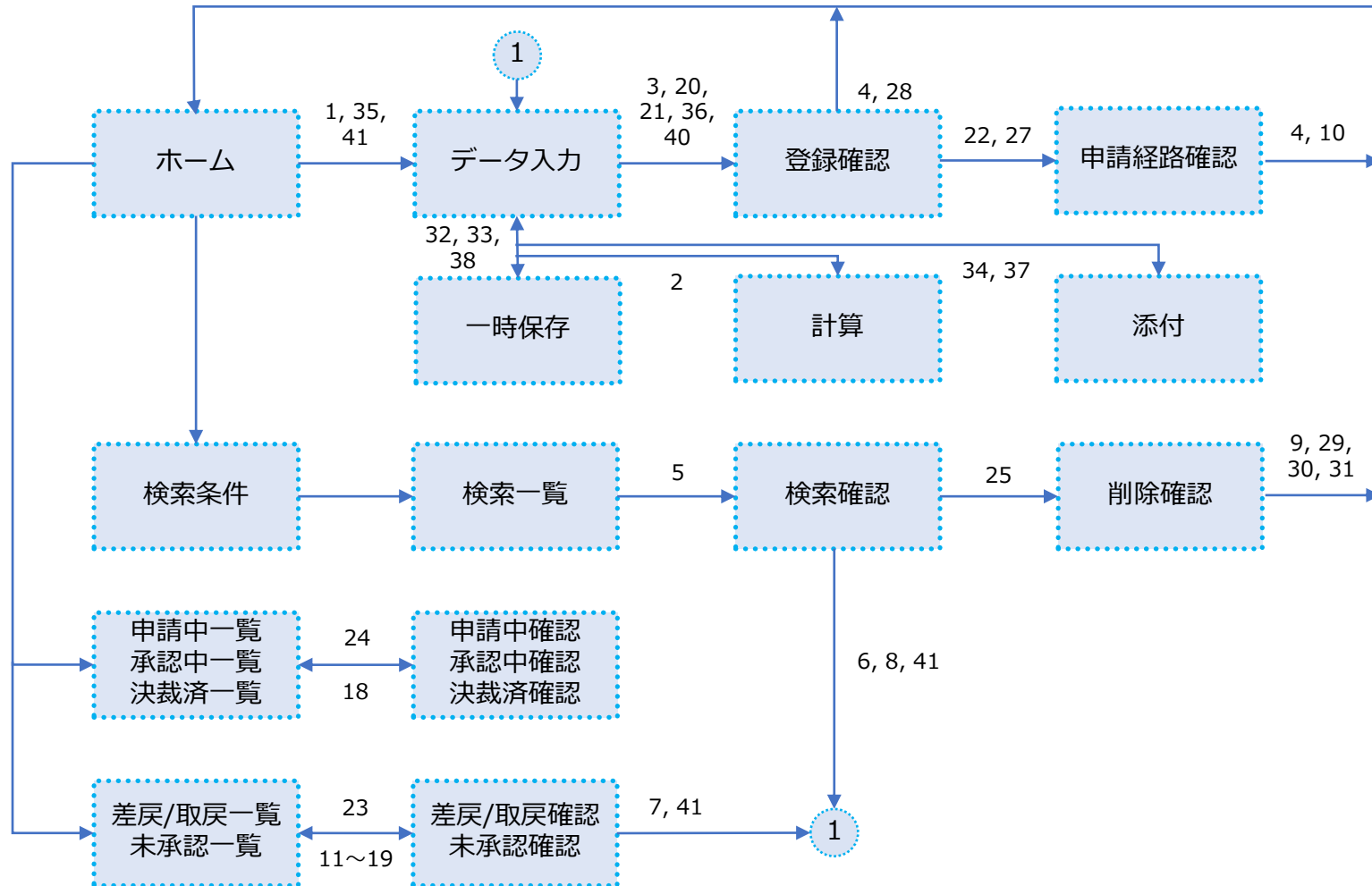
FormPat API は、この開発を可能にするためのカスタマイズモジュール（OptionLib20.dll）のソースプログラムを提供します。C# の開発環境があれば自由にカスタマイズ可能です。

FormPat から エンジンモジュール（eFormLib20.dll）とカスタマイズモジュール（OptionLib20.dll）を参照しているため、OptionLib20.dll から eFormLib20.dll のオブジェクト操作が可能になります。



本書は、FormPat Ver.8.0.0 以降を対象としています。

# カスタマイズ関数の呼び出しタイミング



# カスタマイズ関数の処理一覧（１）

1. アクション処理（モード=1） \*1
2. アクション処理（モード=2）
3. 更新前処理
4. 更新後処理
5. アクション処理（モード=3） \*1
6. アクション処理（モード=4） [修正へ]クリック時 \*1
7. アクション処理（モード=5） \*1
8. アクション処理（モード=6） [参照作成へ]クリック時 \*1
9. アクション処理 削除前処理（モード=7）
10. アクション処理 ワークフロー申請（モード=100） \*2
11. アクション処理 ワークフロー承認（モード=110） \*2
12. アクション処理 ワークフロー却下・再申請請求（申請者へ）（モード=120） \*2
13. アクション処理 ワークフロー却下・再承認請求（前の担当者へ）（モード=130） \*2
14. アクション処理 ワークフロー決裁（モード=140） \*2
15. アクション処理 ワークフロー同報・承認（モード=150） \*2
16. アクション処理 ワークフロー同報・却下（モード=160） \*2
17. アクション処理 ワークフロー申請取消（モード=170） \*2
18. アクション処理 ワークフローズキップ（モード=180） 受信確認・送信確認共通 \*2
19. アクション処理 ワークフロー再申請（モード=190） \*2
20. アクション処理 判定分岐（モード=1000） 3.よりも先に実行
21. アクション処理 登録確認（モード=8） \*1
22. アクション処理 申請経路確認（モード=9） \*1
23. アクション処理 受信確認（モード=10） \*1
24. アクション処理 送信確認（モード=11） \*1
25. アクション処理 削除確認（モード=12） \*1

## カスタマイズ関数の処理一覧（2）

- 26.未使用
- 27.アクション処理（モード=14）[申請へ]クリック時
- 28.アクション処理 登録後処理（モード=30）
- 29.アクション処理 削除後処理（モード=31）
- 30.アクション処理 一時保存データ削除前処理（モード=32）
- 31.アクション処理 一時保存データ削除後処理（モード=33）
- 32.アクション処理 一時保存登録前処理（モード=40）
- 33.アクション処理 一時保存登録後処理（モード=41）
- 34.アクション処理 添付処理（モード=42）
- 35.アクション処理 フォーム表示直後1回のみ（モード=15）
- 36.アクション処理 入力データ退避直後（モード=16）
- 37.アクション処理 入力データ退避直後（モード=17）
- 38.アクション処理 一時保存データ退避直後（モード=18）
- 39.アクション処理 ワークフロー取戻（モード=200）\*2 \*3
- 40.アクション処理 [登録へ]クリック時（モード=19）20.よりも先に実行 \*3
- 41.アクション処理 フォーム表示直後（モード=20）35.の後に実行 \*3

- \*1 フォーム表示サイズ変更・サブウィンドウ表示・[取消][印鑑取消][添付へ]等でポストバックが発生する場合も実行されます。ご注意ください。
- \*2 ワークフロー処理が完了しメール送信後に呼ばれます。
- \*3 Ver.8.2.0 以降の機能です。

# カスタマイズ関数のインターフェース

## 更新前処理

データ入力画面で[登録へ]クリック時に実行されます。

int BeforeUpdate(ref string retstr)

引数 : retstr (出力) ポップアップ表示するメッセージを設定します。

戻り値 : 正常時は0、それ以外はエラーです。

retstr≠""かつ、戻り値≠0ときはポップアップメッセージを表示して処理を中断します。

## 更新後処理

登録確認画面で[登録]クリック時および申請経路確認画面で[申請]クリック時に実行されます。

int AfterUpdate(ref string retstr)

引数 : retstr (出力) ポップアップ表示するメッセージを設定します。

戻り値 : 正常時は0、それ以外はエラーです。

retstr≠""かつ、戻り値≠0ときはポップアップメッセージを表示して処理を中断します。

## アクション処理

各ポイントを通過するときに実行されます。処理は画面遷移の前と後のどちらかになります。

int Action(int mode, ref string retstr)

引数 : mode (入力) アクションモードを取得します。(1~1000)

引数 : retstr (出力) ポップアップ表示するメッセージを設定します。

戻り値 : 正常時は0、それ以外はエラーです。

retstr≠""かつ、戻り値≠0ときはポップアップメッセージを表示し処理は以下の通りです。

- ・ 中断しない : 1, 2, 5~8, 10, 21~25
- ・ 画面遷移のみ中断する : 11~19, 28, 29, 39
- ・ 中断する : 9, 27, 40, 41
- ・ [OK/キャンセル]のメッセージで[キャンセル]選択で中断する : 20

# 共通関数

## 項目値取得

FormPat Designer で定義した項目の入力値を画面から取得します。

```
string f_ItemDataGet(string ItemName)
```

引数 : ItemName (入力) 項目名称を設定します。

戻り値 : 正常時は入力値、それ以外は空白("")です。

## ページ数取得

FormPat Designer で定義したフォームのページ数を取得します。

```
int f_PageCountGet()
```

引数 : なし。

戻り値 : 正常時はページ数、それ以外は0です。

## シート数取得

FormPat の入力画面でページ追加の実行により追加されたページのシート数を取得します。

```
int f_SheetCountGet(int PageNo)
```

引数 : PageNo (入力) ページ番号を設定します。

戻り値 : 正常時はシート数、それ以外は0です。

## 複数シート項目名称取得

複数シートの入力値を取得・設定をするための項目名称を取得します。

```
string f_ItemNameGet(int PageNo, int SheetNo, string ItemName)
```

引数 : PageNo (入力) ページ番号を設定します。

引数 : SheetNo (入力) シート番号を設定します。

引数 : ItemName (入力) 項目名称を設定します。

戻り値 : 正常時は複数シート用項目名称、それ以外は空白("")です。



# 明細行の項目名称

FormPat のデータ入力画面では明細行項目の項目名称は下記のようになります。  
OptionLib20.dll では後ろに「\_[行番号]」を付加した項目名称で値の取得・設定を行います。

FormPat Designer で項目名称「商品名」を明細5行で設計したとき  
OptionLib20.dll では「商品名\_1」「商品名\_2」「商品名\_3」「商品名\_4」「商品名\_5」で値の取得・設定を行います。

## FormPat Designer の設計

項目名称：商品名  
明細行数：5

商品名
商品名
商品名
商品名
商品名



## FormPat のデータ入力画面

商品名_1
商品名_2
商品名_3
商品名_4
商品名_5



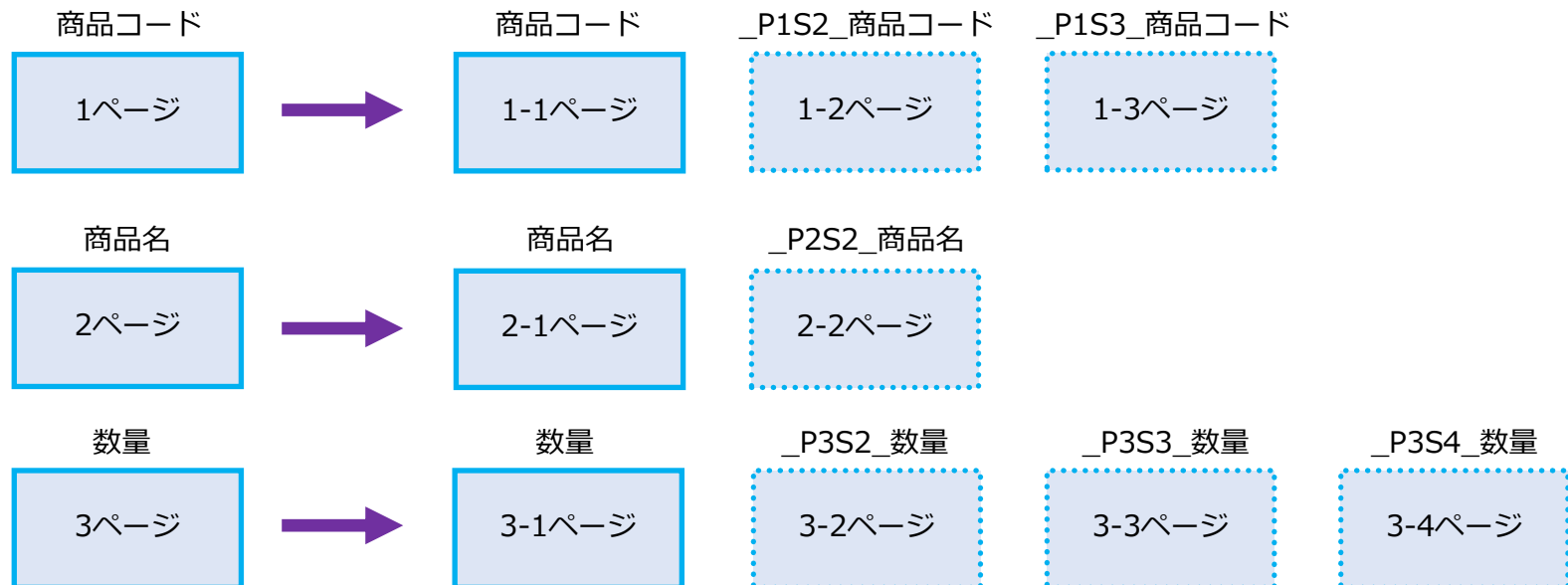
# ページ追加の項目名称

FormPat のデータ入力画面でページ追加の実行により追加されたページの項目名称は下記のようになります。  
OptionLib20.dll では前に「\_P[ページ番号]S[シート番号]\_」を付加した項目名称で値の取得・設定を行います。

FormPat Designer で1ページ目に項目名称「商品コード」、2ページ目に項目名称「商品名」、3ページ目に項目名称「数量」を設計し、FormPat のデータ入力画面で1ページ目を2ページ追加、2ページ目を1ページ追加、3ページ目を3ページ追加したとき OptionLib20.dll では1ページ目の元のページは「商品コード」、追加ページは「\_P1S2\_商品コード」「\_P1S3\_商品コード」で値の取得・設定を行います。同様に2ページ・3ページ目の追加ページも下記のようになります。

## FormPat Designer の設計

## FormPat のデータ入力画面

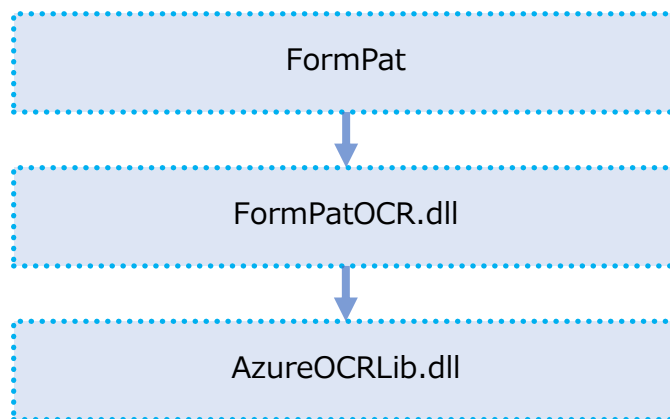


# 光学式文字認識(OCR)の概要

Azure Cognitive Services の Computer Vision と連携して、FormPat の「貼付画像」または「手書き」のポップアップの画像、手書きを挿入時、印刷または手書きのテキストを抽出して FormPat のフォームへ登録できます。\*1 \*2

FormPatOCRライブラリ のモジュール (FormPatOCR.dll) をカスタイズ開発することで、FormPatOCR.dll から AzureOCRライブラリ のモジュール (AzureOCRLib.dll) を利用することで OCR 機能を利用できます。

FormPatOCRライブラリ のモジュール (FormPatOCR.dll) はソースプログラムを提供します。



\*1 Azure Cognitive Services の Computer Vision の契約が必要です。

\*2 Ver.8.4.0 以降の機能です。

# 光学式文字認識(OCR)ライブラリの呼び出しインターフェース

FormPatOCRライブラリ のモジュール (FormPatOCR.dll) は、「貼付画像」または「手書き」ポップアップの登録処理で、FormPat 本体から呼び出される DLL でカスタマイズ開発が可能です。

## ポップアップ処理

画像,手書きポップアップの OCR 処理を行います。

```
int ImageFileToItemValue(int Mode, string ItemName, string FileName,ref object Item, ref object Value, ref string RetStr)
```

引数 : Mode (入力) 処理モード (1 : 貼付画像、2 : 手書き) を取得します。

引数 : ItemName (入力) 項目名称を取得します。

引数 ; FileName (入力) 画像ファイル名を取得します。

引数 : Item (出力) OCR結果を格納する項目名称 (配列形式) を設定します。

引数 : Value (出力) OCR結果の値 (配列形式) を設定します。

引数 : RetStr (出力) 処理結果のメッセージを設定します。

戻り値 : 正常時は0、それ以外はエラーです。

# 光学式文字認識(OCR)ライブラリのインターフェース 1/2

AzureOCRライブラリ のモジュール (AzureOCRLib.dll) は、FormPatOCRライブラリ のモジュール (FormPatOCR.dll) から呼び出される DLL でカスタマイズ開発はできません。

AzureOCRライブラリ のモジュール (AzureOCRLib.dll) は、カスタマイズモジュール (OptionLib20.dll) から利用できます。

## Azure接続キー設定

Azure サービスに接続するためのキーを設定します。

`setAzureKey(string SubscriptionKey, string Endpoint)`

引数 : SubscriptionKey (入力) Azure Cognitive Services のサブスクリプションキーを設定します。

引数 : Endpoint (入力) Azure Cognitive Services エンドポイントを設定します。

## OCR処理

画像から OCR 処理します。

`int ImageFileToText(string FileName)`

引数 : FileName (入力) 画像ファイル名を設定します。

戻り値 : 正常時は0、それ以外はエラーです。

## OCR結果

OCR 結果を取得します。

`string getResultText()`

戻り値 : OCR 結果を JSON 形式で取得します。

## 光学式文字認識(OCR)ライブラリのインターフェース 2/2

### タイムアウト時間設定

OCR 処理のタイムアウト時間を設定します。既定値は1分（60000ミリ秒）です。

`setTimeout(int Timeout)`

引数 : Timeout（入力）タイムアウト時間をミリ秒単位で設定します。

### Webプロキシ設定

デフォルトWebプロキシを使用する場合、プロキシサーバーのアドレスを設定します。

`setDefaultWebProxy(string Address)`

引数 : Address（入力）プロキシサーバーのアドレスを設定します。

### Exception情報取得

エラー発生時のException情報を取得します。

`Exception getException()`

戻り値 : Exceptionクラス

### ■ 参考 ■

- [Microsoft Azure Cognitive Services - Computer Vision ドキュメントサイト ヘリリンク](#)
- OCR結果のJSON仕様は、ドキュメントサイトで[リファレンス]→[OCR REST API v3.2]→[Get Read Result]を参照してください。