

# GEMBA Note for Business 5

## eYACHO for Business 5

### Share for Business 5

# カスタムURLスキーム仕様情報

- Microsoftは、米国Microsoft Corporationの米国およびその他の国における登録商標です
  - Google Driveは、Google Inc.の登録商標です。
  - 本書は株式会社MetaMojが作成したものであり、マニュアルの著作権は、株式会社MetaMojに帰属します。
  - 本書の内容は予告なく変更することがあります。
- 2020年2月27日 ©2020 株式会社MetaMoj

# はじめに

- 本書では、iOS上のほかのアプリケーションと GEMBA Note for Business 5/ eYACHO for Business 5/Share for Business 5との連携をスムーズに行うための仕様を説明しています。
- 連携は、iOS環境におけるカスタムURLスキームの仕組みを利用して行います。
- 本書では、製品名をそれぞれ次のように表記しています。

GEMBA Note for Business 5	GEMBA Note
eYACHO for Business 5	eYACHO
Share for Business 5	Share

# できること

## 1. ほかの（呼び出し元）アプリケーションから取り込む（インポート）

- GEMBA Note/eYACHO/Shareに取り込めるファイル形式
  - PDFファイル (.pdf)
  - 画像ファイル (.png、.gif、.jpg、.jpeg)
  - テキストファイル (.txt)
  - GEMBA/eYACHO 形式 (.dydoc)

## 2. 呼び出し元アプリケーションに送る

- 送ることができるファイル形式
  - PDF ファイル (.pdf) 形式
  - GEMBA/eYACHO 形式 (.dydoc)

## 3. シェアノートを読み出し元アプリケーションに受け渡す（会議開催時）

※Microsoft Office製品のファイルの取り込みにはGoogle DriveのPDF変換機能を利用するため、Google Driveのアカウントが必要になります。

ほかの（呼び出し元）アプリケーションから呼び出す

例） GEMBA/ eYACHO形式のファイルを開く場合

ほかの（呼び出し元）  
アプリケーション  
GEMBA形式 eYACHO形式

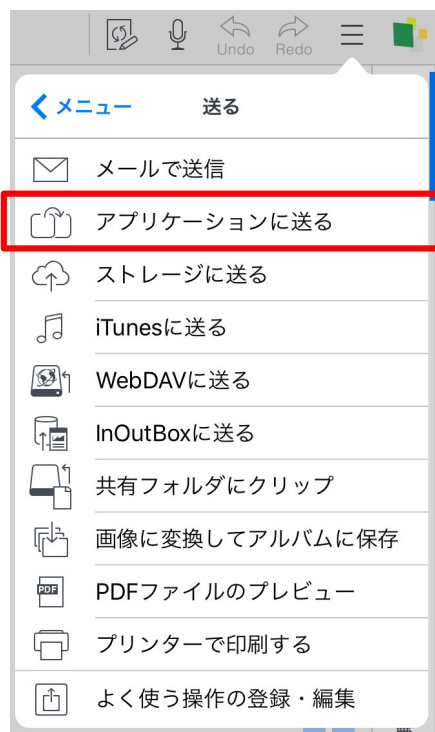
カスタムURLスキーム呼び出し

The screenshot shows an iPad interface for a '不具合管理票' (Defect Management Ticket) application. The form includes fields for '管理番号' (TJ521486), '発行日' (2018/02/14), '検査員' (小泉拓也), '検査工程' (加工), '検査実施時間' (19:30), and '対象ロット番号' (T36254813). A '承認' (Approval) stamp is visible, dated 2018/02/14 and signed by 幸田明美. The '不具合内容' (Defect Content) field contains the text '点検作業中に異音が発生するテキスト' (Text of abnormal sound occurring during inspection work).

Below the form is a diagram titled '図：eYACHO for Business システム構成' (Diagram: eYACHO for Business System Configuration). The diagram illustrates the system architecture, showing components like 'Webブラウザ' (Web Browser), 'eYACHO for Business (クライアントアプリケーション)' (eYACHO for Business (Client Application)), 'インターネット' (Internet), 'A社' (Company A), 'B社' (Company B), 'MetaMojiサーバー' (MetaMoji Server), '管理ツール' (Management Tool), 'ストレージサービス' (Storage Service), and 'シェアサービス' (Share Service).

On the right side of the screenshot, there are two vertical panels. The top panel is labeled 'GEMBA Note' and shows a sidebar with various icons and a '承認' (Approval) stamp. The bottom panel is labeled 'eYACHO' and shows a sidebar with various icons and a '禁止' (Prohibited) sign.

# 呼び出し元アプリケーションに送る



連携オプションに  
"sendbackformat" が指定さ  
れている場合に表示されます

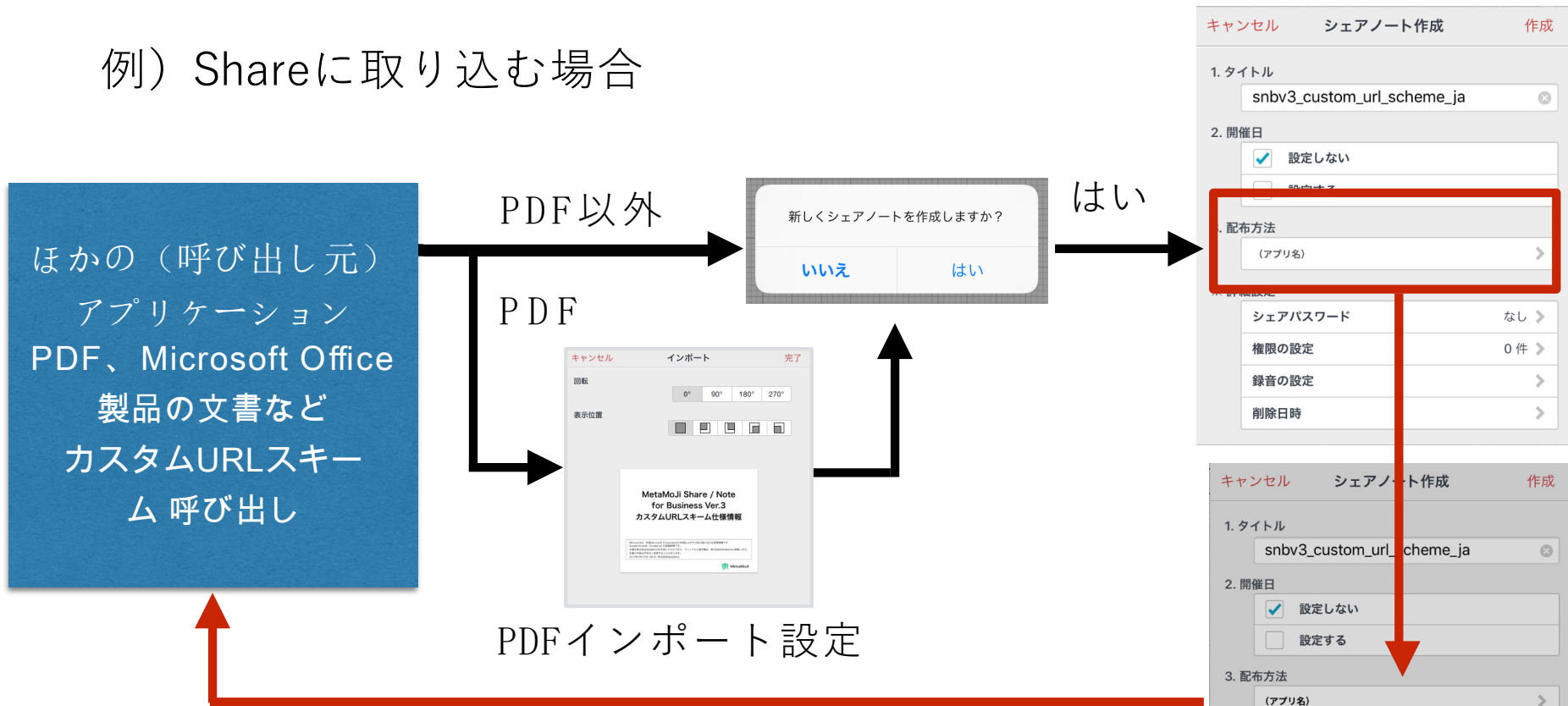
ほかの（呼び出し元）  
アプリケーション

"sendbackformat" オプションで  
指定した形式で送られます  
(PDF / GEMBA / eYACHO形式)

送るメニューに「〇〇（アプリ名）に送る」が表示されるのは、カスタムURLスキームで "sendbackformat" オプションを付けて開いた状態の場合です。  
一度ノートを閉じた（ノート一覧画面に戻った/履歴タブで切り替えた）あと、再度開いた場合は表示されません。

# シェアノートを呼び出し元アプリケーションに受け渡す (会議開催時)

例) Shareに取り込む場合



アプリケーションで配布する選択肢（「○○（アプリ名）で配る」）が表示されるのは、カスタムURLスキームで開いた状態の場合です。一度ノートを開いた（ノート一覧画面に戻った/履歴タブで切り替えた）あと、再度開いた場合は表示されません。

カスタムURLスキームで指定されたアプリケーションで配布する選択肢が表示されます。

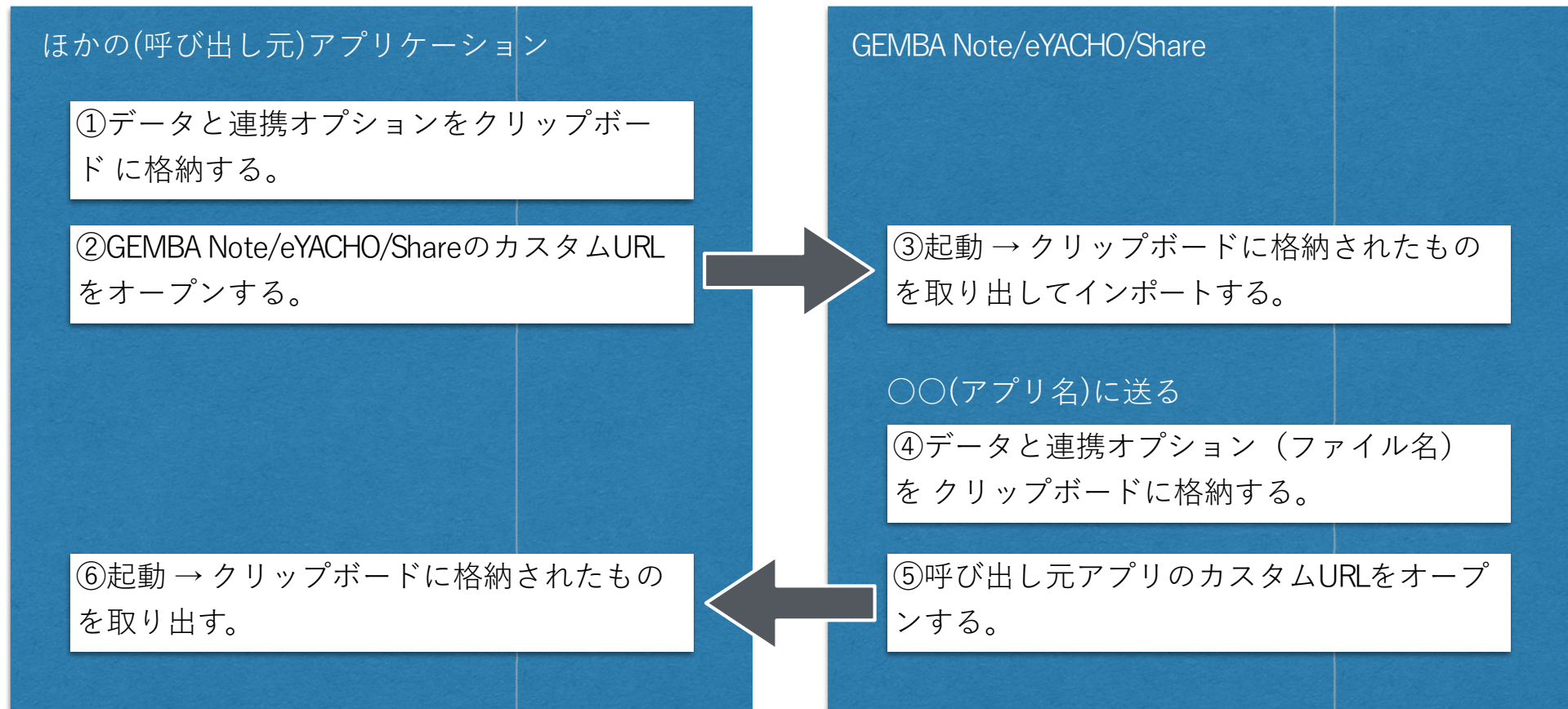
# カスタムURLスキームの仕様

## クリップボードを使った連携

アプリ間で引き渡すデータや連携オプションをクリップボードに格納してから、カスタムURLを使ってアプリを呼び出す連携方法です。

# カスタムURLスキームの仕様

## 1. 連携の流れ





# クリップボードを使った連携

## 2. クリップボードに積む項目

クリップボードの1つ目のアイテムとして、次の2つの表現形式を格納します。

表現形式 (UTI)	型	説明
com.metamoji.noteanytime .importpb.data	Data	PDFやGEMBA/eYACHO形式(.dydoc)などインポートするデータ
com.metamoji.noteanytime .importpb.options	Dictionary	ファイル名や呼び出し元アプリ名などの連携オプションを格納した辞書 ※ 3.連携オプションをご覧ください。

# クリップボードを使った連携

## 3. 連携オプション

キー	型	説明
name	String	ファイル名を想定 ※ ここで指定された拡張子によりデータの扱いが決定されます。
srcname	String	呼び出し元のアプリケーション名 ※ メニューUIなどの文字列として使用されます。
srcurl	String	呼び出し元アプリケーションを呼び出すために使うURL文字列
sendbackformat	String	sendbackformatパラメータを付ける場合は次のいずれかを指定 ・ sendbackformat=dydoc ... GEMBA/eYACHO形式 ・ sendbackformat=pdf ... PDF形式 ※ 必須ではありません。

# クリップボードを使った連携

## 3. 連携オプション

キー	型	説明
nonewshare	Bool	trueを指定すると、インポート後に続けてシェアノートを作成するかどうかを尋ねなくなります。 falseを指定すると従来通りの動きをします(自動的に「はい」が選ばれるわけではありません)。 ※省略時:false
confirmclose	Bool	trueを指定すると、インポートしたノートの編集を終える前に確認メッセージを出すようになります。 falseを指定すると従来通りの動きをします。 ※省略時:false ※srcurlおよびsendbackformatが指定されているときにのみ有効です。

# クリップボードを使った連携

## 4. URLの書式

<スキーム>:///importpb

項目	説明
スキーム	GEMBA Noteで開く : gembanotech5 eYACHOで開く : eyachoch5 Shareで開く : sharech5

# クリップボードを使った連携

## 5. 連携サンプル (Swift 4)

### ◆GEMBA Note/eYACHOの呼び出し

```
func open(_ urlScheme: String, filePath: String, srcName: String, sendBackFormat: String?) {
    guard let data = try? Data(contentsOf: URL(fileURLWithPath: filePath)) else {
        print("Failed to read data!")
        return
    }

    let srcurl = "YOUR-APP:////"

    var options: [String: String] = [:]
    options["name"] = (filePath as NSString).lastPathComponent
    options["srcname"] = srcName
    options["srcurl"] = srcurl
    if let sbformat = sendBackFormat {
        options["sendbackformat"] = sbformat
    }

    let pasteboard = UIPasteboard.general
    guard let optionsPList = try? PropertyListSerialization.data(fromPropertyList: options,
                                                                format: .binary,
                                                                options: 0 ) else {

        print("Failed to make plist!")
        return
    }

    pasteboard.items = [
        [
            "com.metamoji.noteanytime.importpb.data": data,
            "com.metamoji.noteanytime.importpb.options": optionsPList,
        ]
    ]

    let urlString = "¥(urlScheme):////importpb"
    if let url = URL(string: urlString) {
        UIApplication.shared.openURL(url)
    }
}
```

# クリップボードを使った連携

## 5. 連携サンプル (Swift 4)

### ◆GEMBA Note/eYACHOの呼び出し

```
func storeFile() -> Bool {
    let pasteboard = UIPasteboard.general

    guard let data = pasteboard.data(forPasteboardType: "com.metamoji.noteanytime.importpb.data") else { return false }
    guard let optionsData = pasteboard.data(forPasteboardType: "com.metamoji.noteanytime.importpb.options") else { return false }

    guard let optionsAny = try? PropertyListSerialization.propertyList(from: optionsData, options: [], format: nil) else { return false }
    guard let options = optionsAny as? Dictionary<String, String> else { return false }

    guard let file = options["name"] else { return false }

    let documentsPath = NSSearchPathForDirectoriesInDomains(.documentDirectory, .userDomainMask, true)[0]
    let fileURL = URL(fileURLWithPath: file, relativeTo: URL(fileURLWithPath: documentsPath, isDirectory: true))

    return (try? data.write(to: fileURL, options: .atomic)) != nil
}
```

# クリップボードを使った連携

## 5. 連携サンプル (Objective-C)

### ◆GEMBA Note/eYACHOの呼び出し

```
- (void)open:(NSString *)urlScheme withFile:(NSString *)filePath srcName:(NSString *)srcName sendBackFormat:(NSString *)sendBackFormat {
    NSString *srcurl = @"YOUR-APP:///";

    NSMutableDictionary *options = [NSMutableDictionary new];
    options[@"name"] = [filePath lastPathComponent];
    options[@"srcname"] = srcName;
    options[@"srcurl"] = srcurl;
    if (sendBackFormat != nil) {
        options[@"sendbackformat"] = sendBackFormat;
    }
    if (discard) {
        options[@"discard"] = @"1";
    }

    UIPasteboard *pasteboard = [UIPasteboard generalPasteboard];

    pasteboard.items =
    @[
        @{
            @"com.metamoji.noteanytime.importpb.data": [NSData dataWithContentsOfFile:filePath],
            @"com.metamoji.noteanytime.importpb.options": [NSPropertyListSerialization dataWithPropertyList:options
                                                         format:NSPropertyListBinaryFormat_v1_0
                                                         options:0
                                                         error:nil],
        }
    ];

    NSString *urlString = [NSString stringWithFormat:@"%s@:///importpb", urlScheme];
    NSURL *url = [[NSURL alloc] initWithString:urlString];
    UIApplication *app = [UIApplication sharedApplication];
    [app openURL:url];
}
```

# クリップボードを使った連携

## 5. 連携サンプル (Objective-C)

### ◆GEMBA Note/eYACHOの呼び出し

```
- (BOOL)storeFile {
    BOOL result = NO;

    UIPasteboard *pasteboard = [UIPasteboard generalPasteboard];
    NSData *data = [pasteboard dataForPasteboardType:@"com.metamoji.noteanytime.importpb.data"];
    NSData *optionsData = [pasteboard dataForPasteboardType:@"com.metamoji.noteanytime.importpb.options"];
    if (data != nil && optionsData != nil) {
        NSDictionary *options;
        options = [NSPropertyListSerialization propertyListWithData:optionsData options:NSPropertyListImmutable format:nil error:nil];
        if (nil != options && [options isKindOfClass:[NSDictionary class]]) {
            NSString *file = options[@"name"];
            if (nil != file) {
                NSString *documentsPath = [NSSearchPathForDirectoriesInDomains(NSDocumentDirectory, NSUserDomainMask, YES) objectAtIndex:0];
                NSString *filePath = [documentsPath stringByAppendingPathComponent:file];
                [data writeToFile:filePath atomically:YES];
                result = YES;
            }
        }
    }

    return result;
}
```