

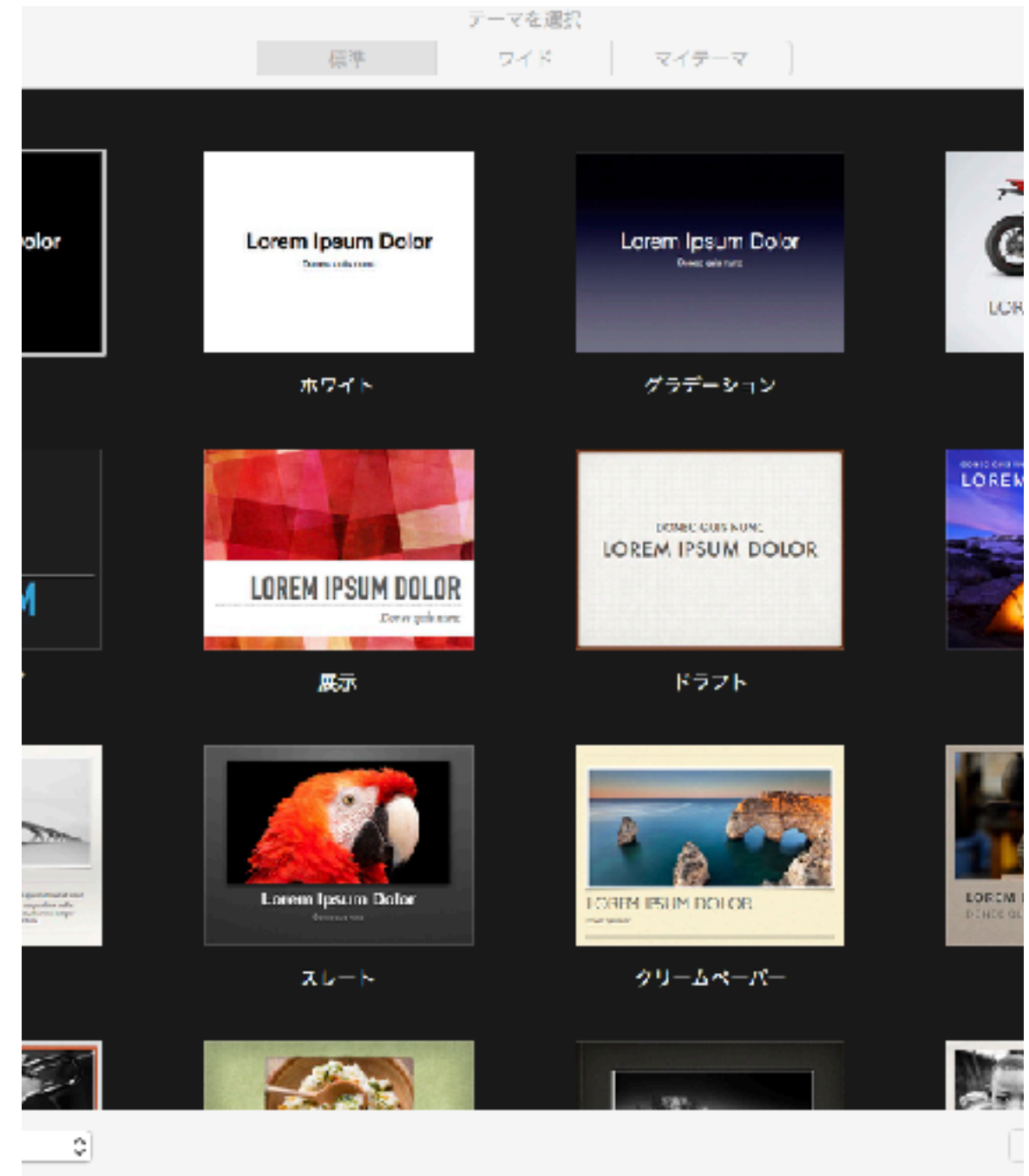
NSDocumentの複数file formatの実装

その3

mindtools@mac.com

目的

- 新規ドキュメントの機能にテンプレートを盛り込みたい



結論

```
open func makeDocument(for urlOrNil: URL?,  
    withContentsOf contentsURL: URL,  
    ofType typeName: String) throws -> NSDocument
```

このメソッドを使うと簡単にできる。

使い方

```
let theDocumentController = NSDocumentController.shared

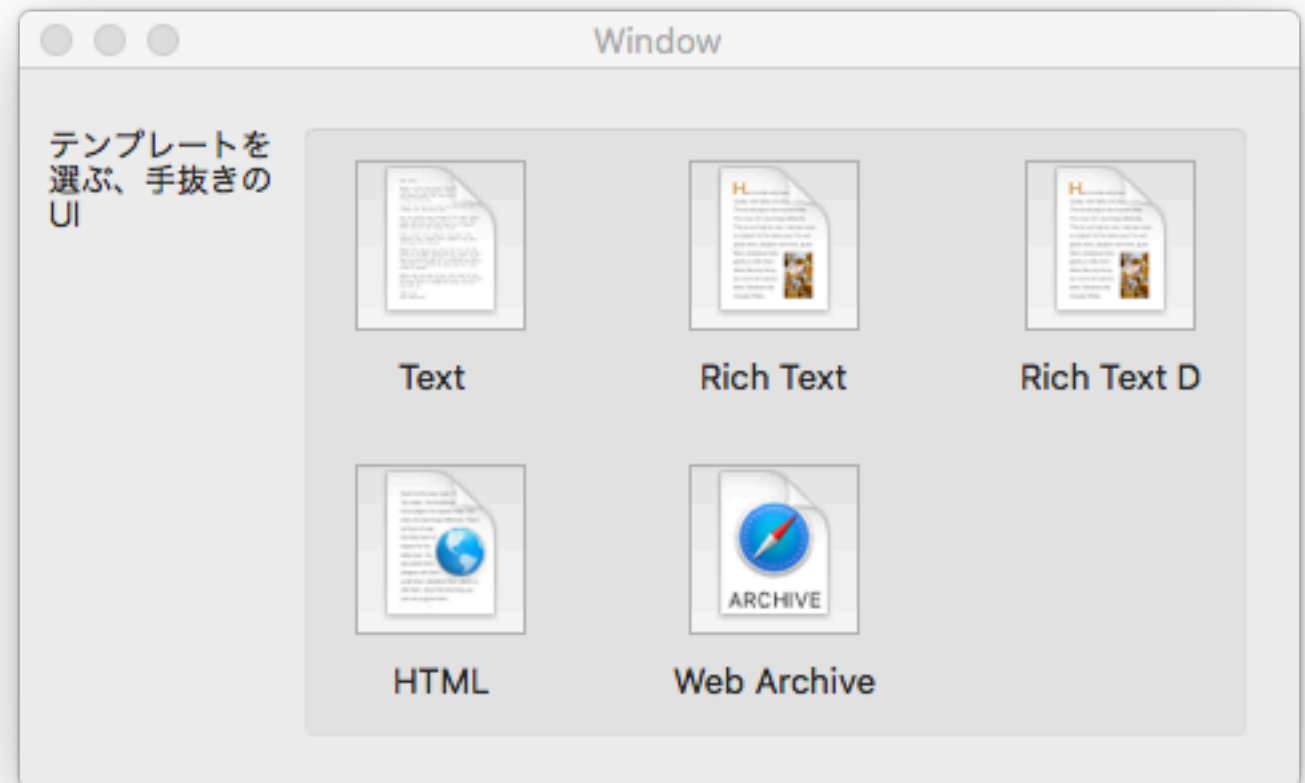
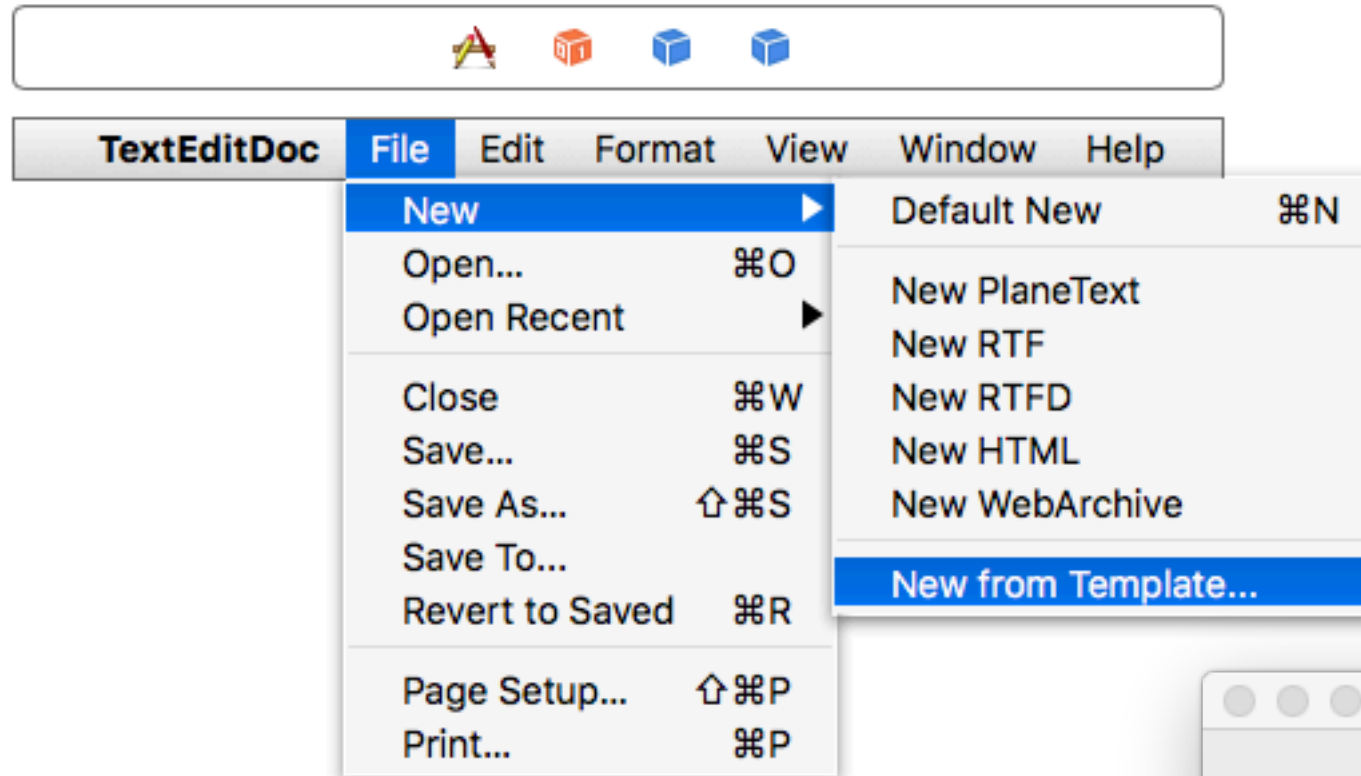
do
{
    let theDocument: NSDocument = try theDocumentController.makeDocument(for: nil ,
                                                                           withContentsOf: contentsURL,
                                                                           ofType: typeName)

    theDocumentController.addDocument(theDocument)
    theDocument.makeWindowControllers()
    theDocument.showWindows()
}
catch
{
    NSError(
        name: NSErrorName.invalidArgumentException,
        reason: "\(typeName) is not a recognized document type.").raise()
}
```

使い方

```
makeDocument(  
    // 新たに作るドキュメントの保存場所。nilは未定  
    for: nil ,  
    // テンプレートファイルを示すURL  
    withContentsOf: contentsURL,  
    // これから作るファイルのファイルタイプ  
    ofType: typeName)
```

コードの見どころ



コードの見どころ

```
class AppDelegate: NSObject, NSApplicationDelegate
{
    .
    .
    .

    @IBAction func newDocumentFromTemplate(_ sender: NSObject)
    {
        // シングルトンのモーダルスダイアログを作り
        // 静的なボタンが複数あり、ボタンを叩くと上記のNSDocumentController.makeDocument(for
        urlOrNil: URL?, withContentsOf contentsURL: URL, ofType typeName: String)で、適当なテンプレートから
        ファイルを作る。

        let theWindowController = TETemplateWindowController.gTemplateWindowControllerInstance
        theWindowController.showWindow(nil)
    }
    .
    .
    .
}
```

- ・ テンプレート一覧ウィンドを作成して表示する

コードの見どころ

```
class TETemplateWindowController: NSWindowController
{
    override var windowNibName: NSNib.Name
    {
        get
        {
            return NSNib.Name("TETemplateWindowController")
        }
    }
}
```

- 同名のxibファイルからwindowを生成する。最近のOSでは書かなくても良いかも。

コードの見どころ

```
class TETemplateWindowController: NSWindowController
{
    @IBAction func newPlaneTextDocumentFromTemplate01(_ sender: NSObject)
    {
        // 閉じる
        self.window?.performClose(nil)

        let theURL: URL = Bundle.main.url(forResource: "PlaneText", withExtension: "txt")!
        self.newDocument(ofType: kUTTypePlainText as String, withContentsOf: theURL)
    }
}
```

- テンプレートウィンドはドキュメント作成後に閉じる。
- バンドルから、テンプレート用のファイルを持ってきてドキュメントを作成

コードの見どころ

```
class TETemplateWindowController: NSWindowController
{
    func newDocument(ofType typeName: String, withContentsOf contentsURL: URL)
    {
        let theDocumentController = NSDocumentController.shared

        do
        {
            let theDocument: NSDocument = try theDocumentController.makeDocument(for: nil ,
                                                                                   withContentsOf: contentsURL,
                                                                                   ofType: typeName)

            theDocumentController.addDocument(theDocument)
            theDocument.makeWindowControllers()
            theDocument.showWindows()
        }
        catch
        {
            NSErrorException(
                name: NSErrorExceptionName.invalidArgumentException,
                reason: "\(typeName) is not a recognized document type.").raise()
        }
    }
}
```

- makeDocument~を使って、documentを作成。
- 後は、定番の処理です。