

RPA学習コース

第4回目：高度なUiPath機能（2）

2021年4月15日 19時

1HB-8S 黄 蔚菁

前回のおさらい

第3回目：高度なUiPath機能（1）

- ▶ エラーハンドリング
- ▶ データテーブルの操作

前回宿題の解説

～～第三回目宿題の解説 & 質問コーナー～～

宿題内容：

サイト「<https://info.finance.yahoo.co.jp/ranking/?kd=1&tm=d&mk=1>」をアクセスし、**東証1部に上場する銘柄**の中から、**出来高先頭50位**の銘柄情報を抽出して、CSVに保存してください。抽出してほしい情報は「**銘柄コード、銘柄名称、前日終値、PBR**」となる。

※前日終値とPBRは、銘柄詳細画面にある。

前回宿題の解説

マーケット関連ランキング

値上がり率
値下がり率
ストップ高
ストップ安
年初来高値更新
年初来安値更新
出来高
単元当り出来高
出来高増加率
出来高減少率
売買代金上位
売買代金下位
時価総額上位

出来高 東証1部 デイリー

最終更新日時:2021年4月1日16時09分

1~50件/2191件中

順位	コード	市場	名称	取引値	前日比	出来高	揭示
1	8306	東証1部	(株)三菱UFJフィナンシャル・グループ	15:00 595.1	+0.57% +3.4	79,224,700	揭示
2	8604	東証1部	野村ホールディングス(株)	15:00 582.7	+0.22% +1.3	38,945,300	揭示
3	8918	東証1部	(株)ランド	15:00 11	+10.00% +1	31,358,900	揭示
4	5020	東証1部	ENEOSホールディングス(株)	15:00 493.6	-1.59% -8	22,886,800	揭示
5	7211	東証1部	三菱自動車(株)	15:00 304	-3.49% -11	22,801,400	揭示
6	7201	東証1部	日産自動車(株)	15:00 590.7	-4.09% -25.2	22,262,900	揭示
7	8698	東証1部	東証1部	15:00 887	+0.55% +33	19,953,800	揭示
8	4689	東証1部	東証1部	15:00 19,608	+12.5%	19,608,100	揭示
9	8750	東証1部	第一生命ホールディングス(株)	15:00 2,083	+9.52% +181	18,737,600	揭示

銘柄コードをクリックしたら右のページに遷移

銘柄一覧画面

8306 銀行業 リアルタイム株価 15:00
(株)三菱UFJフィナンシャル・グループ **595.1**

ポートフォリオに追加

詳細情報 チャート | 時系列 | ニュース | 企業情報 | 掲示板 | 株主優待 | レポート | 業績予想 3 | みんかぶ 3



前日終値

PBR

前回宿題の解説

空の結果データテーブルを構築

銘柄一覧画面のすべて情報をデータテーブルに格納

データ行ごとに繰り返し

銘柄コードから銘柄詳細のURLを組み立てる

銘柄詳細のURLをアクセスし、前日終値、BPRを取得

空のデータ行に銘柄コード、名称、前日終値、BPRを格納

上記データ行を結果データテーブルに追加

結果データテーブルをCSVに出力

前回宿題の解説

詳細画面に辿り着く方法

固定文字列

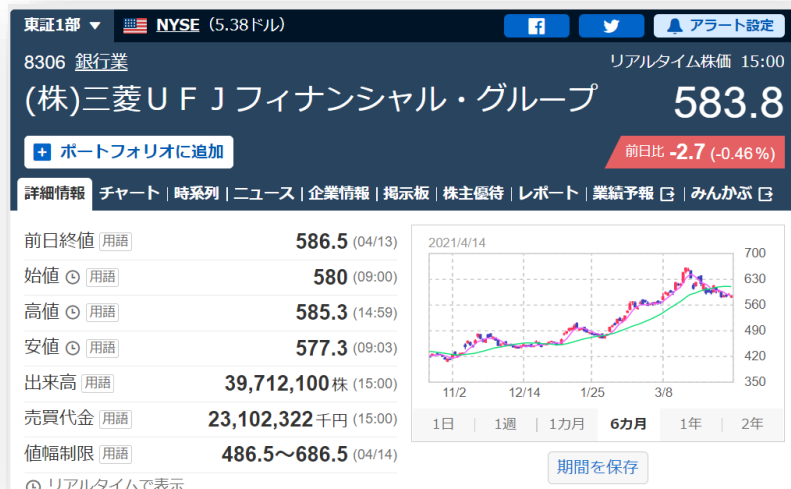
銘柄コードで変わる部分



本日の勉強会を終えると、あなたは...

- ▶ 3rdパーティUiPathライブラリでロボ機能を拡張できる
- ▶ UiPathライブラリを開発できる
- ▶ C#（もしくはVB.net）でカスタマイズ機能を開発できる（C#知識が必要）
- ▶ 今日の授業を終えると、あなたはこんなロボットを作れる！

銘柄コードを入力



読み上げ：
三菱UFJフィナンシャル・グループ
の株価は538.8円です！

アジェンダ

- ▶ UiPathライブラリ
- ▶ カスタマイズコーディング (C#で実現)
- ▶ 宿題

UiPathライブラリ

ライブラリについて

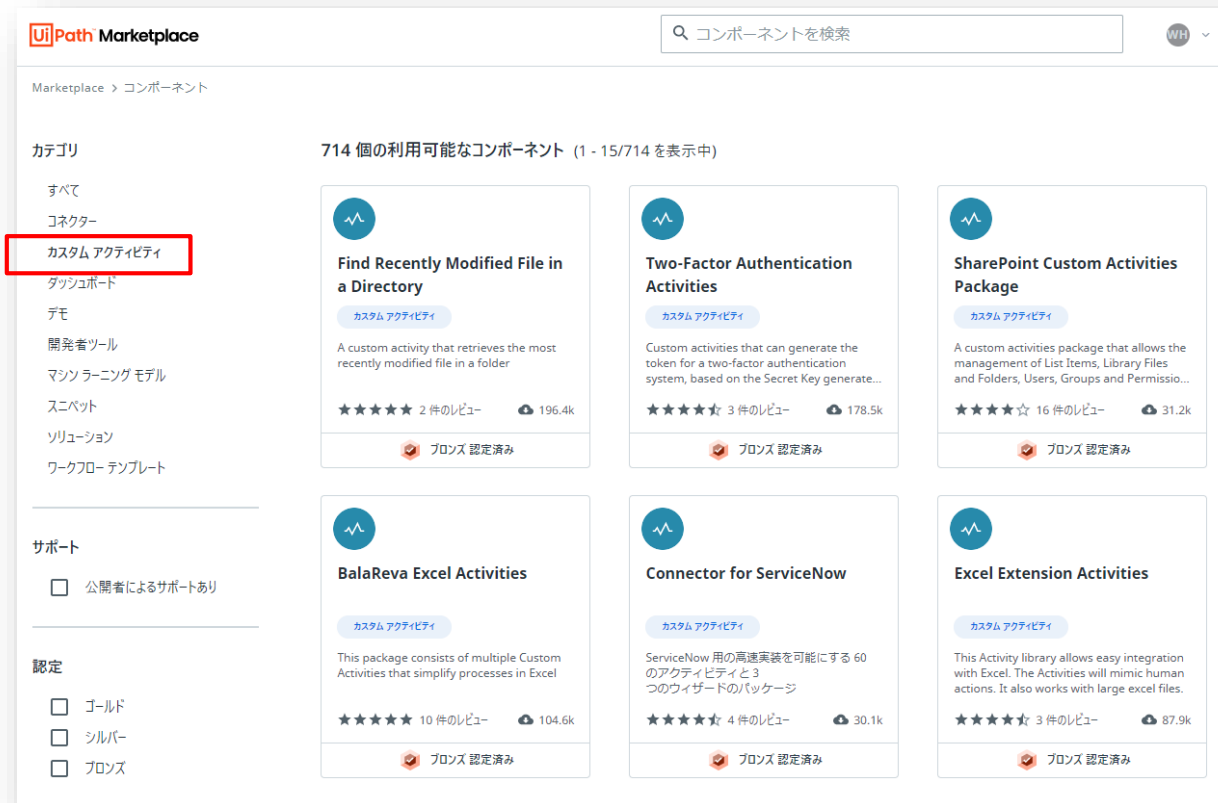
- ▶ ライブラリとは、複数の再利用可能なコンポーネントを含むパッケージをいう。ライブラリをプロジェクトに追加することにより、利用可能なアクティビティを増やすことができ、UiPathに本来なかった機能を簡単に使える。



UiPathライブラリ

UiPath Marketplace

▶ <https://marketplace.uipath.com/ja>



■ マーケットプレイス
ユーザや公式が作成した部品(ア
クティビティやワークフロー)等
を共有するというサービス

UiPathライブラリ

3rdパーティUiPathライブラリの追加

プロジェクト

プロジェクトを検索 (Ctrl+Alt+P)

読み上げライブラリ

依存関係

右クリック

管理

2.10.3-preview

UiPath.Mail.Activities = 1.10.3-preview

UiPath.System.Activities = 21.4.0-preview

UiPath.UIAutomation.Activities = 21.4.1-preview

インティティ

.settings

Main.xaml

project.json

パッケージを管理

設定

プロジェクト依存関係

すべてのパッケージ

ローカル

オフィシャル

マーケットプレイス

json

フィルター: プレリリースを含む

UiPathTeam.JsonRequestActivities 作成者: Nadia Malij v1.0.0
A HTTP Request activity that accepts a Json request as input.

Auxiliobits.DataTableToJson.Activities 作成者: Sandeej v1.0.0
This Component convert datatable into json string format

UiPathTeam.JsonDataConverter.Activities 作成者: Rahi v1.1.0
A package containing 3 custom activities for JSON/JSONArray to other data formats: to CSV file

Json.NET 作成者: James Newton-King v12.0.3
Json.NET is a popular high-performance JSON framework for .NET

UiPathTeam.JSONUtilities.Activities 作成者: tuhin.sam v0.5.5
This Package contains following activities:
1. Convert Datable to JSON
2. Convert JSON to CSV

インストール

バージョン: 1.0.0

ランタイム ルール: ①
ストリクト

説明:
This Component convert datatable into json string format

バージョン: 1.0.0

作成者: Sandeep Pathak

ライセンス: [ライセンス情報を表示](#)

公開日: 2020/10/29 6:56:51

タグ: Json,web,Web,Data ,Processing

保存

キャンセル

インストールして保存すると、該当ライブラリがプロジェクトの依存関係に追加される

マーケットプレイスを選択

UiPathライブラリ

3rdパーティUiPathライブラリの利用

ライブラリIDがクリップボードコピーされる
パッケージ管理画面に入力して検索すると該当ライブラリを追加できる

The image shows a screenshot of the UiPath Marketplace and the 'Manage Packages' dialog in UiPath Studio. The Marketplace page for 'Auxiliobits - DataTable To JSON String' is displayed, showing a gold certification badge, a custom activity label, a 4.1k rating, and a '概要' (Overview) tab. The 'Manage Packages' dialog in the foreground shows a search box with the text 'Paste Here (in search box)' and a list of packages. The packages listed are 'UiPath.Salesforce.Activities' (v1.5.2) and 'UiPath.Script.Activities' (v1.0.7495.12930). A red callout bubble points to the copy icon next to the package name in the Marketplace, indicating that the library ID can be copied to the clipboard.

Auxiliobits - DataTable To JSON String

ゴールド 認定済み

カスタム アクティビティ

★★★★★ 4 件のレビュー 4.1k

概要 レビュー 4 質問

要約

This Custom Activity converts DataTable to JSON

クリックして **Auxiliobits.DataTableToJson.Activities** をクリップボードにコピーします。
このパッケージ名をコピーして UiPath Studio 内の [パッケージを管理] ダイアログに簡単に貼り付けられるようにします。

Manage Packages

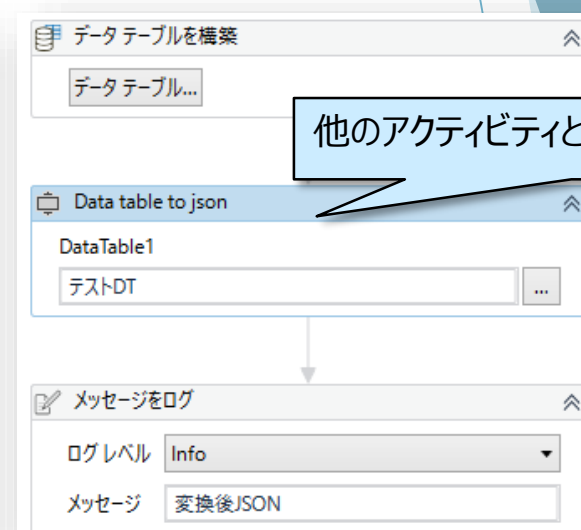
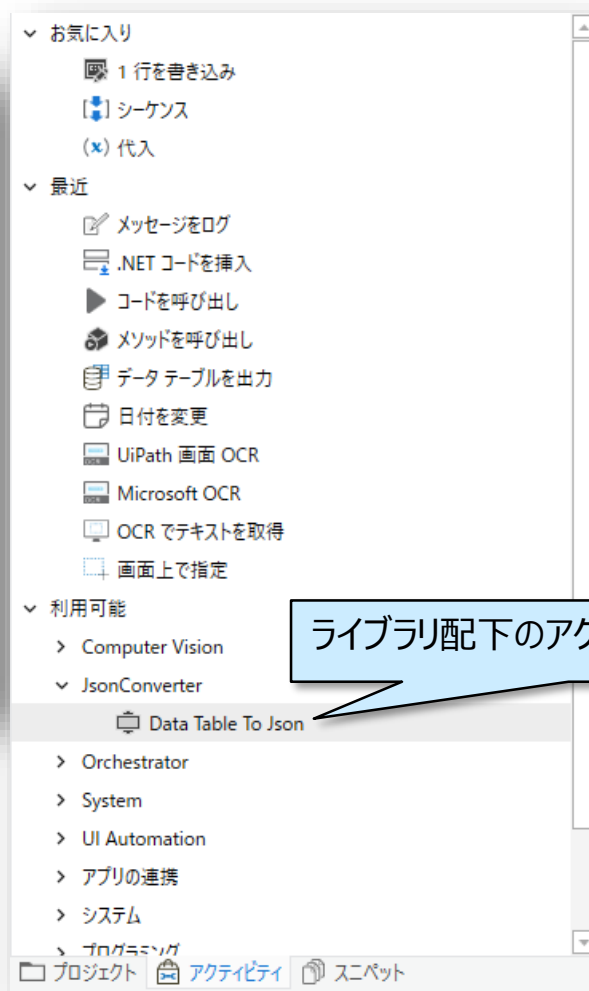
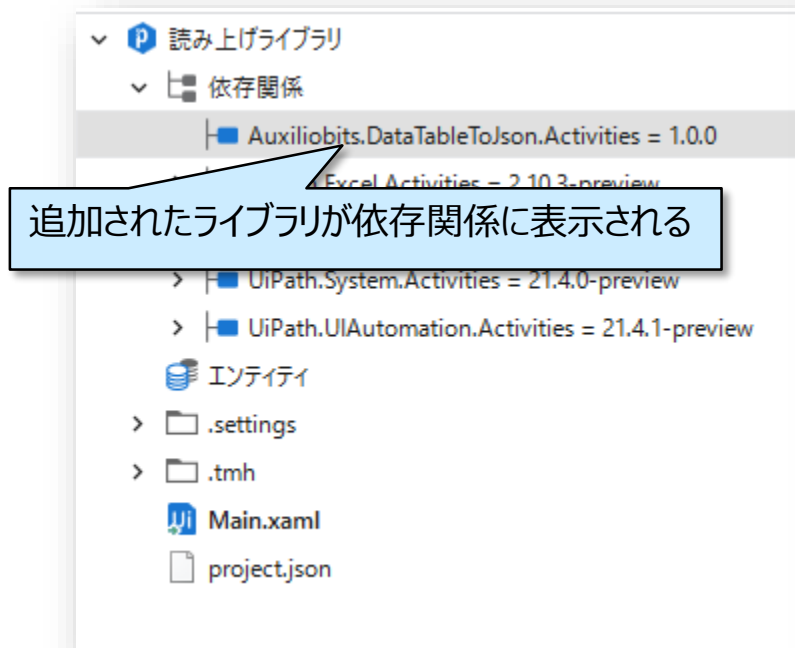
Paste Here (in search box)

Filter: None

Package Name	Version
UiPath.Salesforce.Activities by UiPath	v1.5.2
UiPath.Script.Activities by UiPath	v1.0.7495.12930

UiPathライブラリ

3rdパーティUiPathライブラリの利用



UiPathライブラリ

練習時間

練習時間（10分）

～～ライブラリの利用練習～～

サイト「<https://info.finance.yahoo.co.jp/ranking/?kd=1&tm=d&mk=1>」をアクセスし、画面の銘柄情報をデータテーブルに抽出して、JSONフォーマットに変換したうえ、ログに出力してください。

※データテーブルからJSONへの変換は下記ライブラリを利用してください。

<https://marketplace.uipath.com/ja/listings/datatable-to-json-string>

UiPathライブラリ

ライブラリ化するメリット

- ▶ 1つのワークフローで取り扱う問題を小さくする。
- ▶ 重複する処理を集約して、同一のライブラリとして部品化する。
- ▶ ライブラリ単位でテストができるようになる。
- ▶ チーム開発を容易にする。
- ▶ 今後の開発で流用できる。

以前紹介したサブワークフロー化のメリットをライブラリにもすべてある

UiPathライブラリ

ライブラリを開発する

▶ ライブラリとサブワークフローの違い

	ライブラリ	サブワークフロー
複数のUiPathプロジェクト間の流用性	そのまま流用可能	コピーペすれば流用可能
依存管理	されている	されない
バージョン管理	されている	されない

UiPathライブラリ

ライブラリを開発する

開く

閉じる

スタート

テンプレート

チーム

ツール

設定

ヘルプ

UiPath

Studio Pro 2021.4.0
Community License
EXE インストーラー

UiPath Studio Pro コミュニティ

開く

 ローカル プロジェクトを開く
移動して既存のプロジェクトを開きます。

 複製またはチェックアウト
ソース管理リポジトリ (例: Git, TFS, SVN) から開きます。

最近使用したプロジェクトを開く

 読み上げライブラリ
空のプロジェクト

 CSharpモジュール
CSharpモジュール

 OCR黄テスト
空のプロジェクト

 OCRサンプル
空のプロジェクト

 メモ帳2倍
空のプロジェクト

 第三回目宿題
空のプロジェクト

 第二回目練習用ロボット
空のプロジェクト

 第三回目サンプル
株価取得 (データテーブルサンプル)

 WEBSERVE自動入力
エクセル表を読み込動急入力を自動化する

 空のプロセス1
空のプロジェクト

新規プロジェクト

 プロセス
空のプロジェクトで始めて、新しい自動化プロセスをデザインします。

 ライブラリ
再利用可能なコンポーネントを作成して、まとめてライブラリとしてパブリッシュします。ライブラリは自動化プロセスに依存関係として追加できます。

 テストオートメーション
空のプロジェクトで始めて、新しいテストプロジェクトを作成します。

 テンプレート
他のオートメーションプロジェクトで使用できるテンプレートを作成します。

テンプレートから新規作成

 Robotic Enterprise Framework
大規模なデプロイにおけるベストプラクティスに従ったランザクションビジネスプロセスを作成します。

 オークストレーションプロセス
サービスのオークストレーション、人間の介入、および実行時間の長いランザクションを使用するプロセスを実装します。

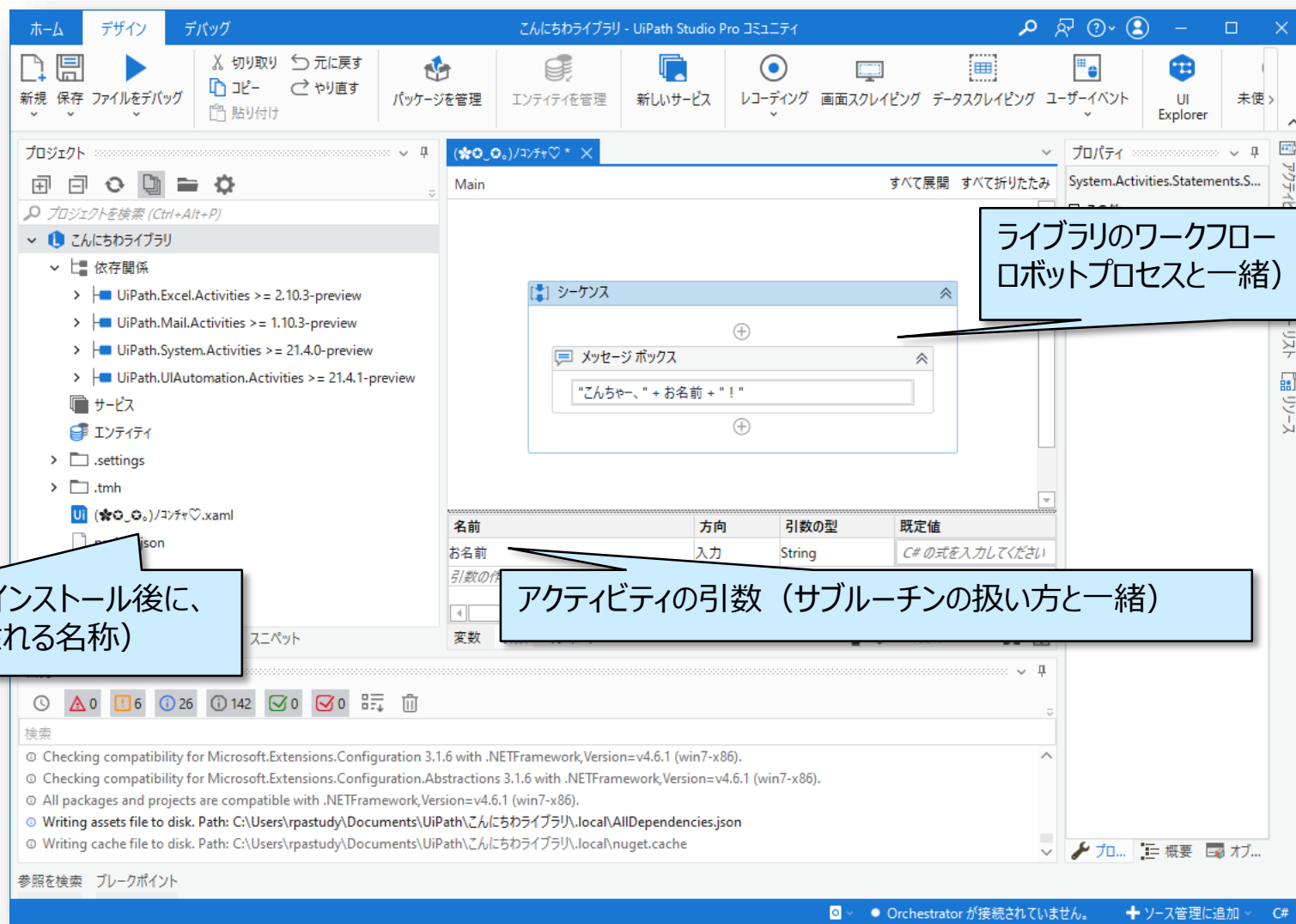
 トランザクションプロセス
プロセスをフローチャートダイアグラムとしてモデル化します。

 トリガーベースの有人のプロセス
マウスまたはキーボード操作によるユーザーイベントに反応

その他のテンプレート

UiPathライブラリ

例：こんにちわライブラリ



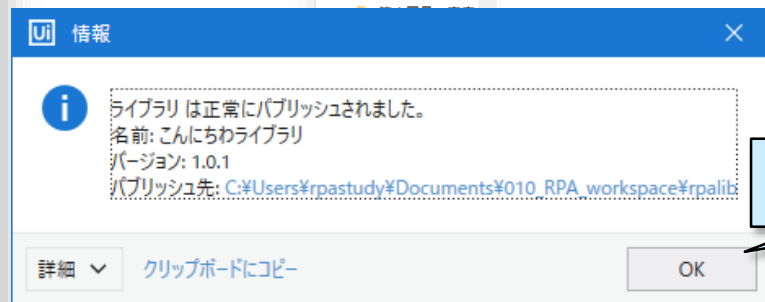
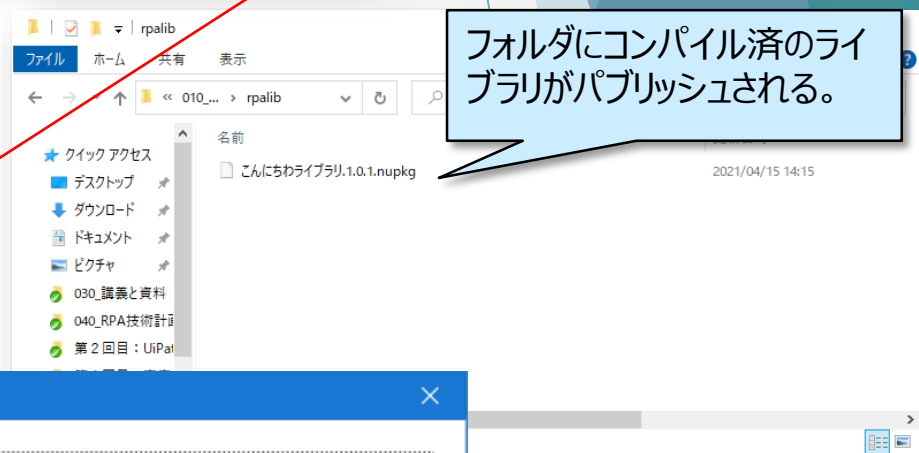
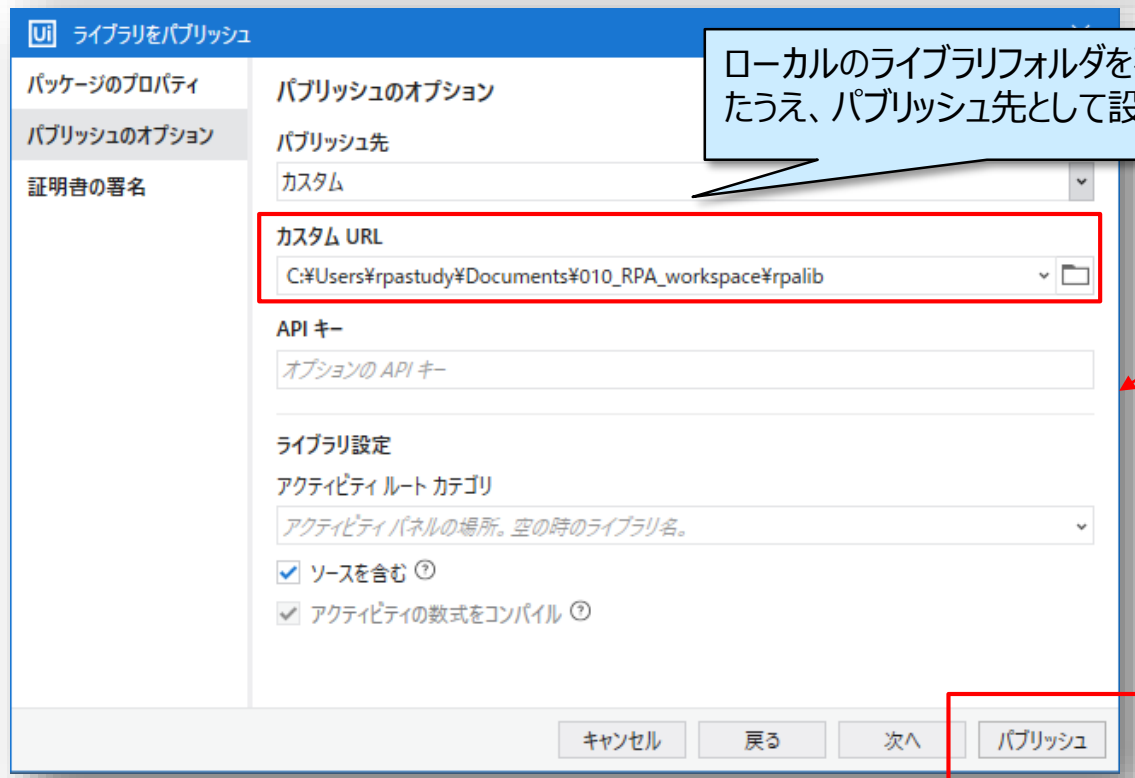
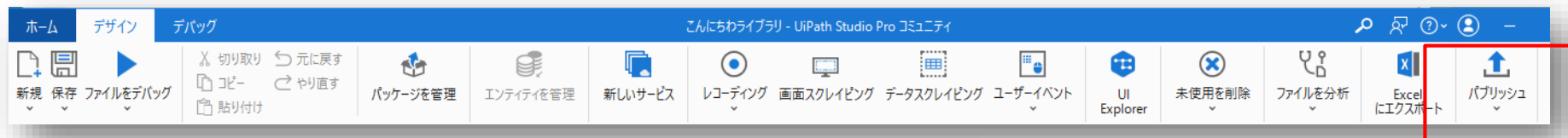
アクティビティ名 (ライブラリインストール後に、アクティビティパネルで表示される名称)

ライブラリのワークフロー (開発方法は通常のロボットプロセスと一緒に)

アクティビティの引数 (サブルーチンの扱い方と一緒に)

UiPathライブラリ

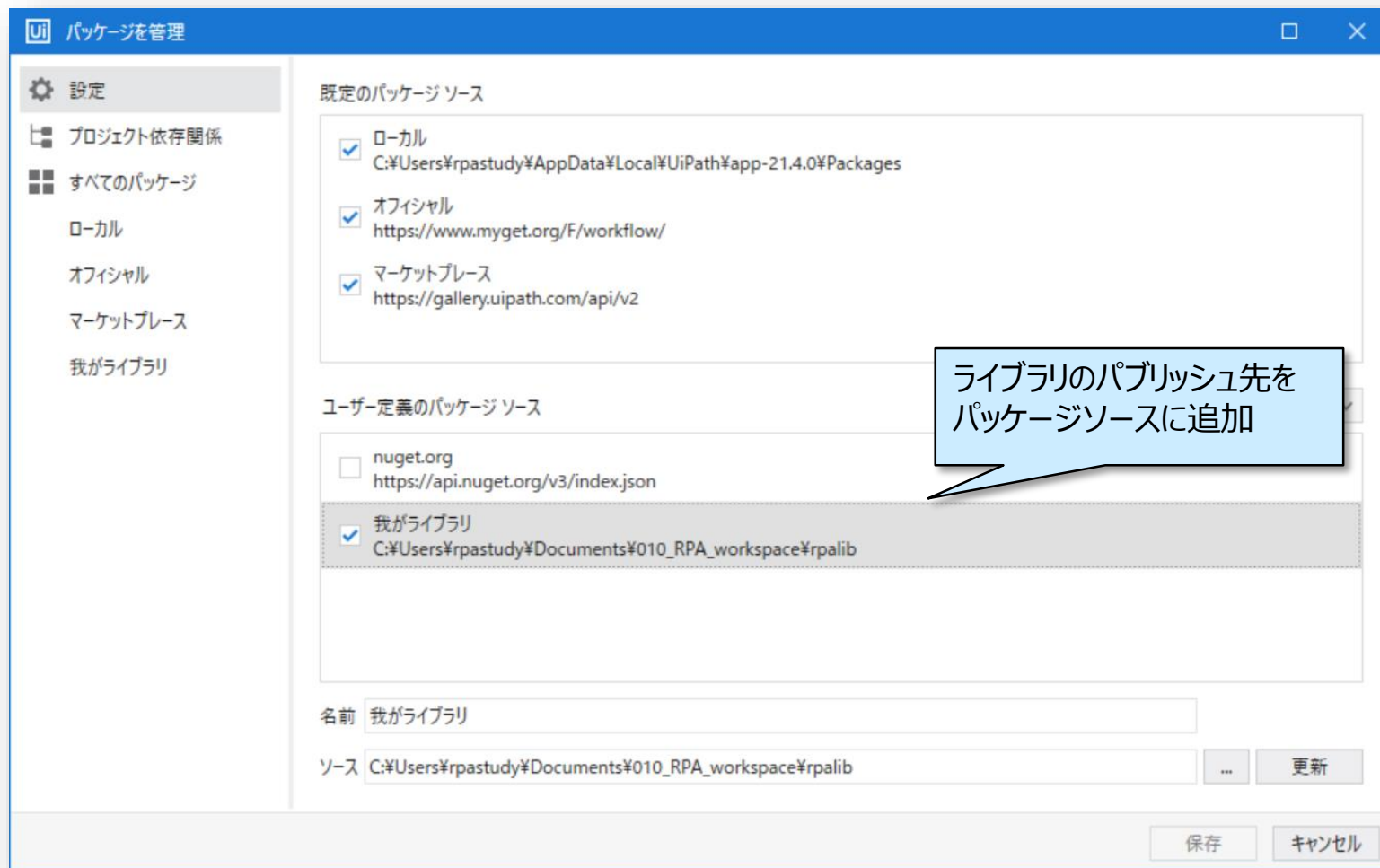
ライブラリのパブリッシュ



パブリッシュ完了！

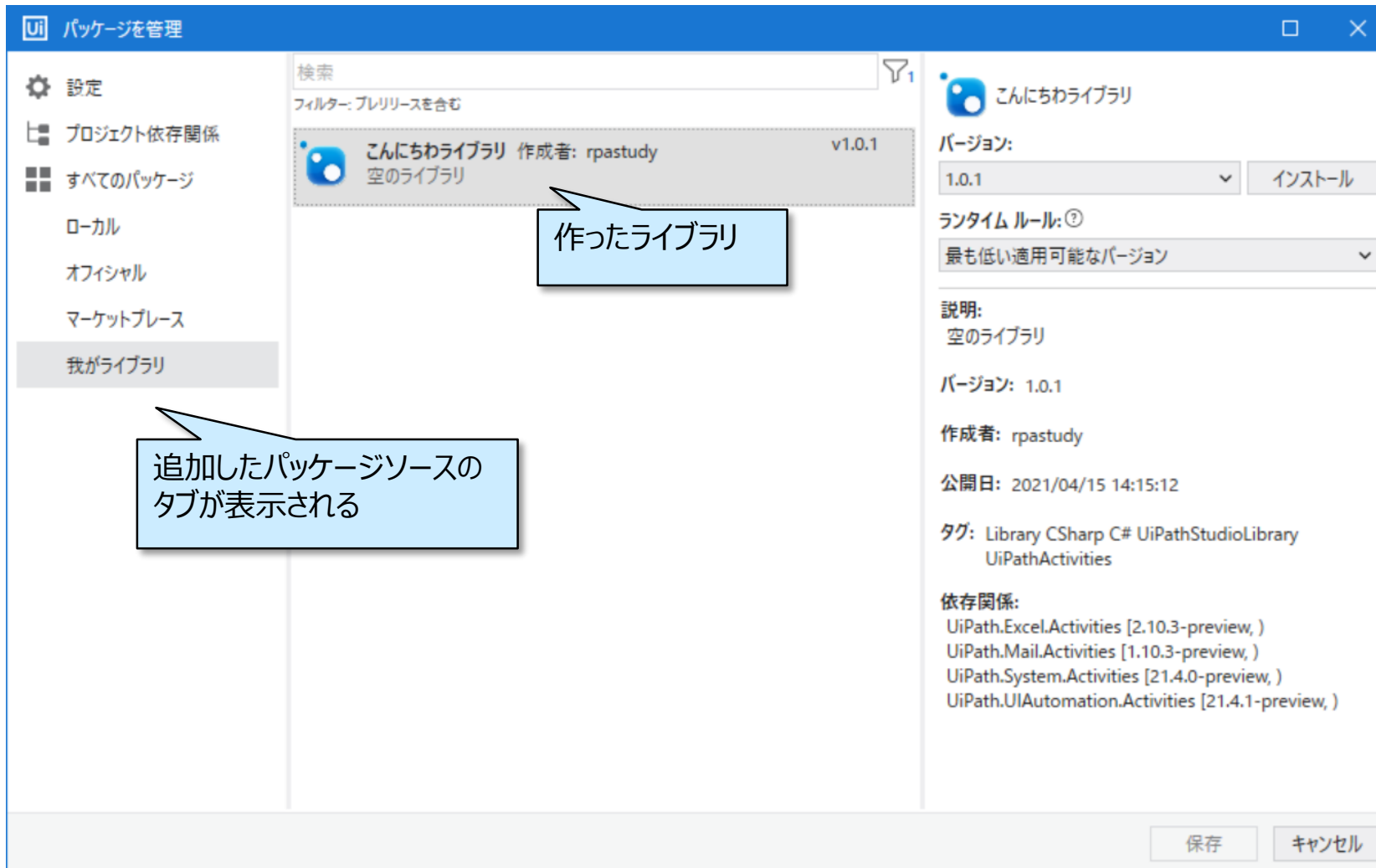
UiPathライブラリ

開発したライブラリを利用：パッケージソースの追加



UiPathライブラリ

開発したライブラリを利用：自作ライブラリを選択



UiPathライブラリ

開発したライブラリを利用：自作ライブラリを使う

The screenshot displays the UiPath Studio Pro interface. On the left, the 'Library' pane is open, showing a list of available and installed activities. A callout box points to the custom activity '(*)_コンチャ♡' with the text 'アクティビティファイル名で表示される' (Displayed by activity file name). The main workspace shows a workflow with a 'Sequence' container containing the custom activity. A callout box points to the 'Arguments' property of the activity with the text '引数' (Arguments). A 'Message Box' dialog is shown with the text '黄さん、こんちゃー！' and an 'OK' button. A callout box points to the dialog with the text '実行結果' (Execution result). The bottom status bar indicates 'Orchestrator が接続されていません。' (Orchestrator is not connected.) and 'ソース管理に追加' (Add to source control).

利用可能

- > Computer Vision
- > Orchestrator
- > System
- > UI Automation
- > アプリの連携
- ▼ こんにちわライブラリ
 - (*)_コンチャ♡
- > システム
- > プログラミング
- > ユーザー イベント
- > ワークフロー

アクティビティファイル名で表示される

引数

実行結果

黄さん、こんちゃー！

OK

Orchestrator が接続されていません。 + ソース管理に追加 C#

UiPathライブラリ

練習時間

練習時間（10分）

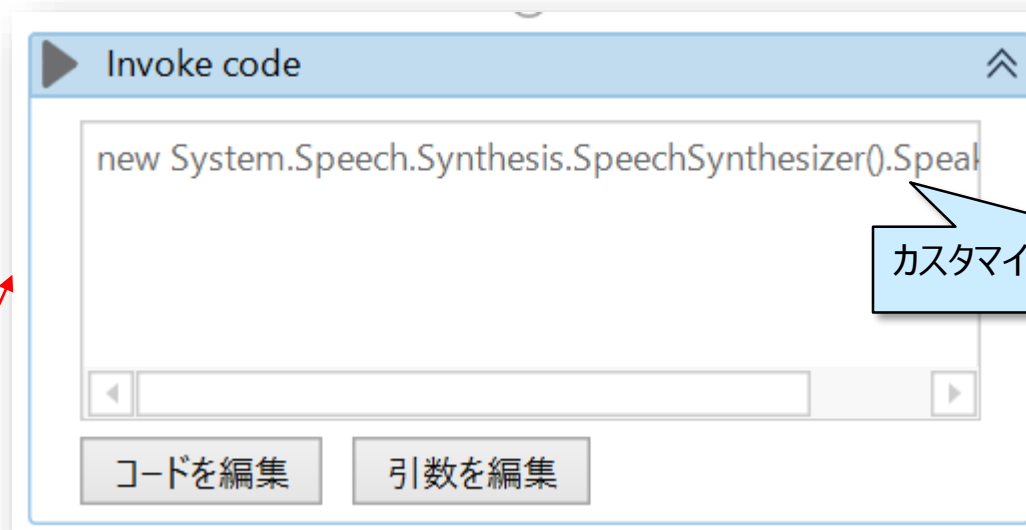
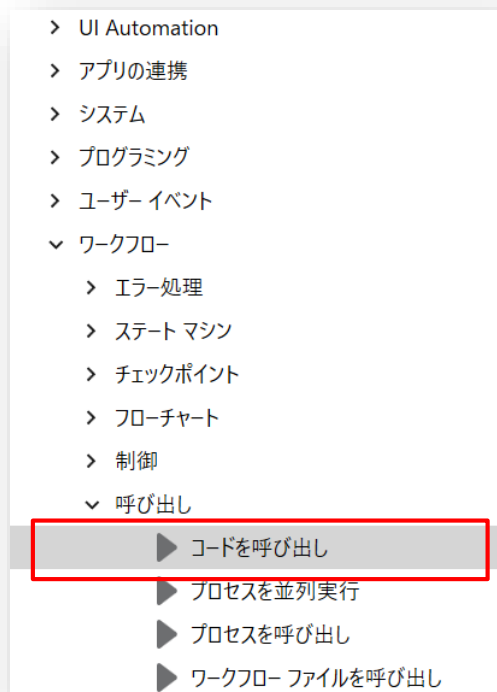
～～ライブラリの開発練習～～

サンプルの「こんにちわライブラリ」を作って、ロボットプロセスから呼び出してください。

カスタマイズコーディング

「コード呼び出し」アクティビティについて

- ▶ 「コード呼び出し」アクティビティを利用して、C#もしくはVB.netで書いたコードをロボットプロセスに組み込むことができる。「コード呼び出し」をうまく利用することで、より安定的な、強いロボットを作れる。



カスタマイズコードを組み込める

カスタマイズコーディング

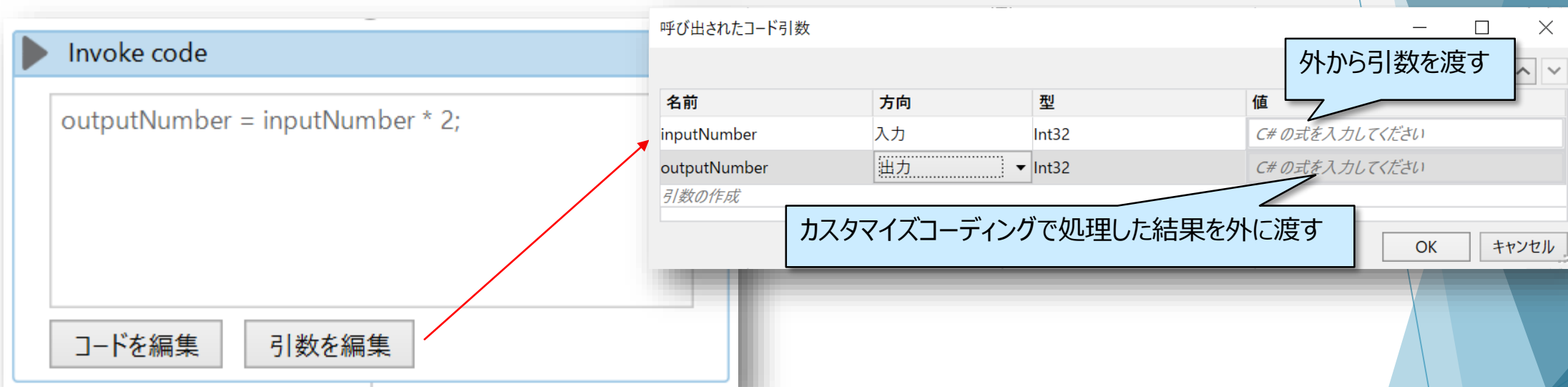
なぜカスタマイズコーディングが必要か

比較項目	UI操作系のRPA (UiPath、...)	伝統的なプログラミング言語 (Java、C#、...)
開発ハードルの高さ	やや低い 業務担当者自らでも開発可能 注：入門が簡単だが、メンテナンス性、拡張性、安定性の高いロボットを開発するのに、右記の知識が必要になってくる。	高い コーディング、ソフトウェア いとまずは作れない
移植しやすさ	移植がやや面倒 稼働環境が開発環境との一致性（ソフト、解像度、設定など）を保たないとリリース後に動かない可能性が高い	移植しやすい JavaならJVM、.NETならCLRがあればとにかく動く
UI操作	得意 UI操作のために生まれた技術なので、得意なのは当然	普通はやらない .netならマウスイベントなどを発生させることもできるのだが、開発がとても難しい
スピード	遅い 人間のようにUI操作で処理するため、人間の操作と比べ数倍だけ早い。それに実行中にUIを占有するため、二重実行が不可。	高速 CPUとメモリの世界で動くため、RPAの数万倍も速い。マルチスレッド実行も可能。

1回目の勉強会でお見せした比較表

カスタマイズコーディング

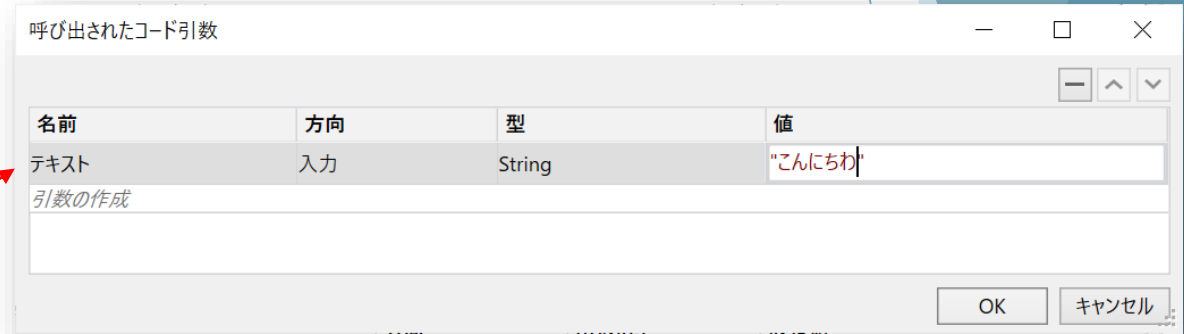
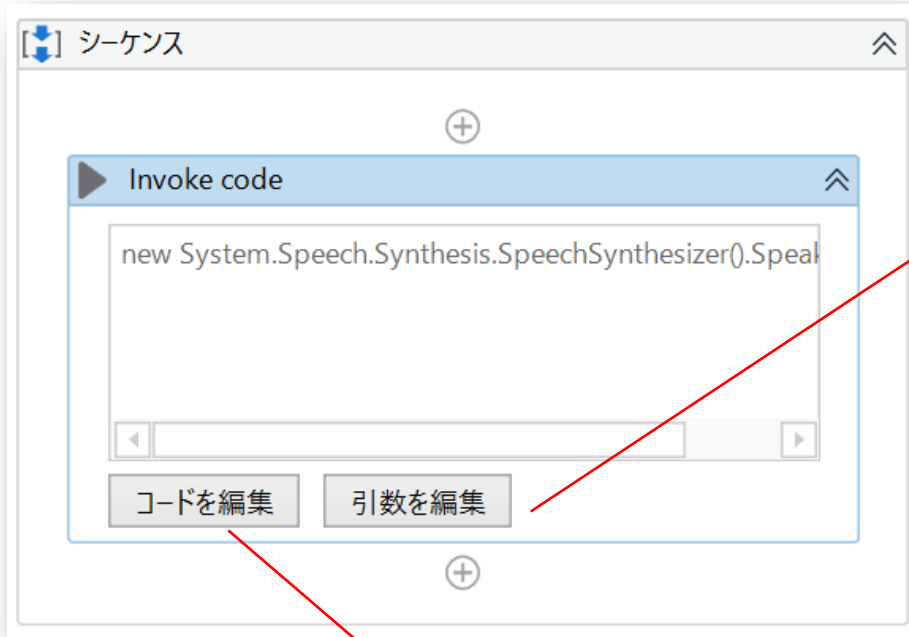
サンプル①：数字二倍化



入力した数字を二倍にして返すカスタマイズ処理

カスタマイズコーディング

サンプル②：おしゃべり



渡された引数（テキスト）の内容を読み上げる

```
new System.Speech.Synthesis.SpeechSynthesizer().Speak(テキスト);
```

音声合成のライブラリはUiPathのデフォルトについてないため、次のページに従い、追加する必要がある。

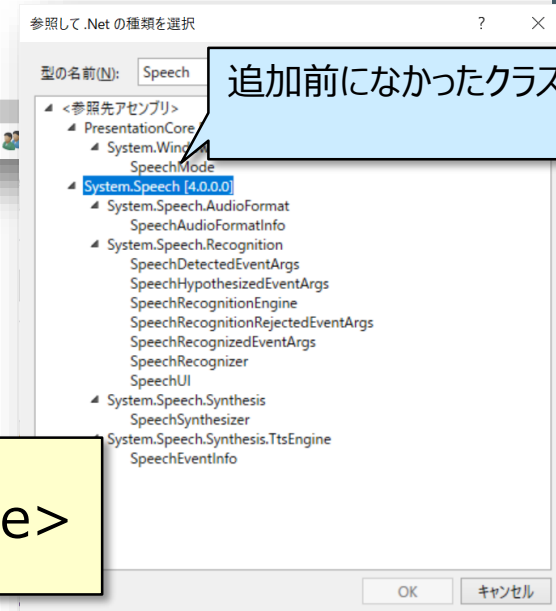
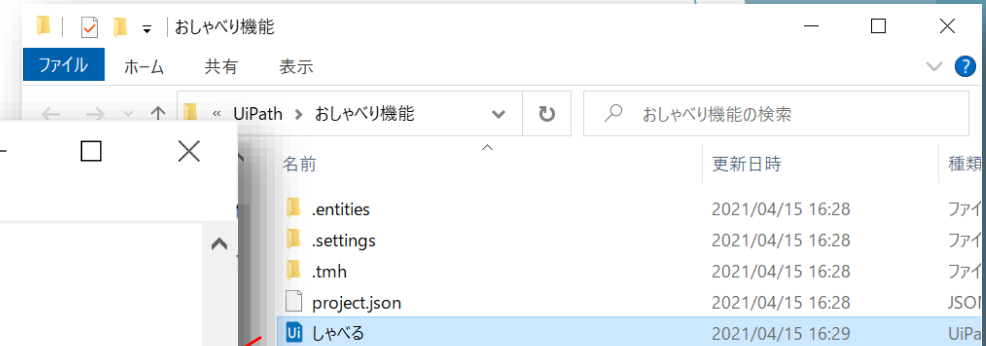
カスタマイズコーディング

.netリファレンスの追加

UiPathプロジェクト以外のデフォルト.netライブラリを参照する場合、下記の追加手順が必要

```
<AssemblyReference>System.Data</AssemblyReference>
<AssemblyReference>System</AssemblyReference>
<AssemblyReference>System.Drawing</AssemblyReference>
<AssemblyReference>System.Core</AssemblyReference>
<AssemblyReference>System.Xml</AssemblyReference>
<AssemblyReference>System.Xml.Linq</AssemblyReference>
<AssemblyReference>PresentationFramework</AssemblyReference>
<AssemblyReference>WindowsBase</AssemblyReference>
<AssemblyReference>PresentationCore</AssemblyReference>
<AssemblyReference>System.Xaml</AssemblyReference>
<AssemblyReference>UiPath.System.Activities</AssemblyReference>
<AssemblyReference>UiPath.UiAutomation.Activities</AssemblyReference>
<AssemblyReference>System.Speech</AssemblyReference>
</sco:Collection>
</TextExpression.ReferencesForImplementation>
</Activity>
```

音声合成ライブラリを追加する。
追加後、アクティビティファイルを開きなおしてください。



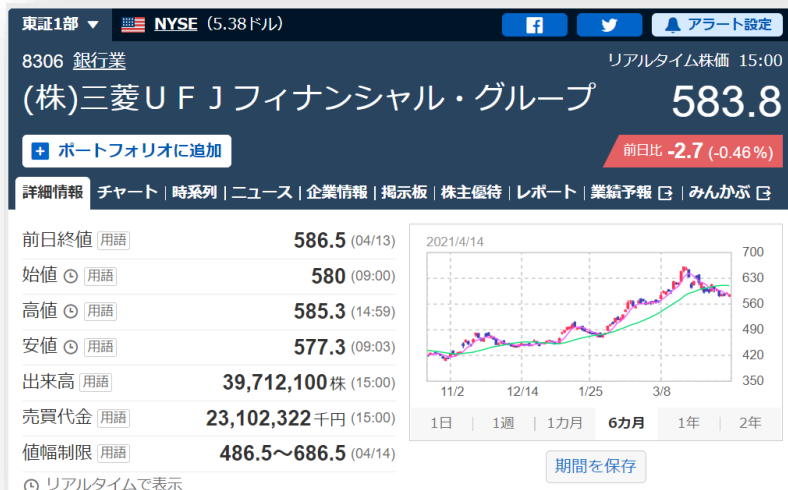
追加前になかったクラスが増えた

<AssemblyReference>System.Speech</AssemblyReference>

カスタマイズコーディング

株価の読み上げ

銘柄コードを入力



読み上げ：
三菱UFJフィナンシャル・グループ
の株価は538.8円です！

質問コーナー

宿題

～～株価読み上げロボ～～

カスタマイズコーディングの株価読み上げロボットを完成してください。

■ 処理内容

- ①「おしゃべりライブラリ」を完成してパブリッシュする。
- ②ロボットプロセスの利用者に銘柄コードを入力してもらう。
- ③銘柄コードで下記サイトから銘柄名称と株価を取得する。
サイト「<https://finance.yahoo.co.jp/quote/4481.T>」
- ④[銘柄名] + “の株価は” + 「株価」 + “です！”の文字列を組み立てる。
- ⑤上記文字列を①の「おしゃべりライブラリ」で読み上げる。

次回予告

第5回目：高度なUiPath機能（3）（5/6（木） 19時）

- ▶ リモートデスクトップ処理の自動化
- ▶ デバッグ方法の紹介
- ▶ Gitでロボットのバージョン管理を行う

4月29じゃない、ご注意！