

JP1 × Power Automate for desktop 連携ガイド

(第 3 版)

2023 年 8 月

株式会社 日立製作所
JP1-RPA 支援センター

変更内容

第 3 版での変更内容

追加・変更内容	変更箇所
3 種類の連携方法全体を説明するため、2 章を追加	2
デスクトップ連携(個人業務の自動化モデル)を 3 章に追加	3
サーバーデスクトップ連携(共通業務の自動化/一元管理モデル)を 4 章に記載	4
Azure 連携(クラウド部門業務の標準化/自動化モデル)を 5 章に記載	5

第 2 版での変更内容

追加・変更内容	変更箇所
オンプレミス データ ゲートウェイを利用しないフロー実行の追加に伴い、以下の内容を追加・変更 概要図に記載の Power Automate サービスから運用管理・ロボット実行端末へのプロセス実行線の追加 JP1/AJS3 と Power Automate for desktop との連携に関する前提条件の追加 「表 5.3 前提ソフトウェア一覧」からオンプレミス データ ゲートウェイを削除 3.2 章の冒頭に「オンプレミス データ ゲートウェイを利用した場合の手順」であることを明記 オンプレミス データ ゲートウェイを利用しないフロー実行のための手順を追加	5.1.1、5.1.3、 5.2.1、5.2.2、 5.2.3
Power Automate のクラウドフローから、 SharePoint Online チームサイトへのファイル作成と JP1 サーバーから SharePoint Online チームサイトへの ファイル参照・削除の説明を追記	5.1.1、5.1.3、 5.2.3、5.2.6
SharePoint Online を構成に追記	5.1.3
前提製品に SharePoint Online Client Components SDK を追記	5.2.1
ツールのディレクトリ・ファイル構成を変更	5.2.4
Power Automate for desktop 連携ツールの内容を追記	5.2.5
指定するユーザの「多要素認証の設定」に関する説明を追記	5.2.6
ツールの実行結果コードを追加	5.3.1.2

はじめに

本ガイドは、「JP1」と「Power Automate for desktop」との連携方法を明らかにすることで、業務自動化の範囲を広げ、運用の効率向上を図ることを目的としています。

■対象読者

本ガイドは、次の知識をお持ちの方にお読みいただくことを前提としています。

- ・ Power Automate for desktop に関する基本的な機能や操作
- ・ Windows のバッチプログラムの作成

■お願い事項

本ガイドに記載の情報は、2023年8月末時点の特定環境における連携の検証結果を、事例としてご紹介するものです。

- ・可能な限り正確な内容を記載するよう努めていますが、その内容を保証するものではありません。
 - ・記載された内容によって生じた損害等の一切の責任を負いかねますので、ご了承ください。
 - ・画面表示をはじめ、製品仕様は、改良のため変更することがあります。
- ご利用に際しては、お客さま、パートナーさまご自身でご検証のうえ、ご判断いただけますようお願いいたします。

■輸出時の注意

本製品を輸出される場合には、外国為替及び外国貿易法の規制ならびに米国の輸出管理規則など外国の輸出関連法規をご確認のうえ、必要な手続きをお取りください。

なお、ご不明な場合は、当社担当営業にお問い合わせください。

■他社商品名、商標などの引用に関する表示

Azure、Power Automate、PowerShell および Windows は、マイクロソフト 企業グループの商標です。

その他記載の会社名、製品名などは、それぞれの会社の商標もしくは登録商標です。

■著作権

Copyright (C) 2021, 2023, Hitachi, Ltd.

Copyright (C) 2021, 2023, Hitachi Solutions, Ltd.

目次

1. 製品の総称および略称	7
2. 本連携ガイドで説明する連携方法について	8
3. デスクトップ連携(個人業務の自動化モデル)	9
3.1. 概要	9
3.1.1. JP1/CPA と PAD 連携概要	9
3.1.2. 前提プログラム	10
3.1.3. 関連マニュアル	10
3.1.4. システム構成	10
3.2. JP1/CPA と Power Automate for desktop の連携方法	11
3.2.1. Power Automate for desktop のデスクトップフローを JP1/CPA から実行するために必要な Power Automate for desktop の設定	11
3.2.2. Power Automate for desktop のデスクトップフロー実行に必要な JP1/CPA の設定	11
3.2.2.1. JP1/CPA のアクションフローの作成	12
3.2.2.2. Power Automate for desktop 実行ユニットの作成	13
3.2.2.3. JP1/CPA のアクションフローの登録	17
3.2.3. Power Automate for desktop のデスクトップフロー実行	18
3.2.3.1. ジョブの作成	18
3.2.3.2. ジョブの実行	19
3.2.3.3. ジョブの実行結果確認	20
3.3. Power Automate for desktop で実行完了ファイルを使用して実行結果を判断するサンプル	21
4. サーバーデスクトップ連携(共通業務の自動化/一元管理モデル)	31
5. Azure 連携(クラウド部門業務の標準化/自動化モデル)	32
5.1. 概要	32
5.1.1. Power Automate for desktop 連携ツール 概要	32
5.1.2. Power Automate for desktop 連携ツール メリットおよび注意事項	33
5.1.3. JP1/AJS3 と Power Automate for desktop との連携に関する前提条件	33
5.2. Power Automate for desktop と JP1/AJS3 の連携方法	35
5.2.1. 前提条件	35
5.2.2. Power Automate for desktop 環境設定手順	36
5.2.2.1. Power Automate for desktop 環境での、Microsoft アカウント設定方法	36
5.2.3. Power Automate クラウドフロー作成手順	46
5.2.3.1. Power Automate for desktop を実行するための、Power Automate クラウドフ	

ロー作成手順	46
5.2.3.2. Power Automate クラウドフローHTTP 要求受信時トリガーURL の取得	60
5.2.4. Power Automate for desktop 連携ツールの構成	61
5.2.5. Power Automate for desktop 連携ツール インストール手順	62
5.2.5.1. Power Automate for desktop 連携ツールの配置	62
5.2.6. Power Automate for desktop 連携ツール セットアップ手順	69
5.2.6.1. Power Automate for desktop 連携ツール 構成定義	69
5.2.7. Power Automate for desktop 連携ツール アンインストール手順	71
5.2.7.1. Power Automate for desktop 連携ツールの削除	71
5.3. 稼働確認	72
5.3.1. Power Automate for desktop 連携ツールを利用した JP1/AJS3 と Power Automate for desktop との連携方法	72
5.3.1.1. 実行	72
5.3.1.2. 結果確認	73
5.4. トラブルシューティング	76
5.4.1. 異常時の対処	76

1. 製品の総称および略称

本ガイドで使用する製品の総称および略称を次に示します。

表 1.1 製品総称

総称	説明
JP1/Automatic Job Management System 3	RPA 製品が実行する業務を含め、サーバーで行う基幹業務の実行スケジュールや実行順序を制御する製品
JP1/Client Process Automation	RPA 製品が実行する業務を含め、PC で行う定型業務の実行スケジュールや実行順序を制御する製品

表 1.2 製品略称

略称	正式名称
JP1/AJS3	JP1/Automatic Job Management System 3
JP1/AJS3 - Agent	JP1/Automatic Job Management System 3 - Agent
JP1/AJS3 - Manager	JP1/Automatic Job Management System 3 - Manager
JP1/AJS3 - View	JP1/Automatic Job Management System 3 - View
JP1/CPA	JP1/Client Process Automation

表 1.3 マイクロソフト製品の表記

略称	正式名称
Windows 10	Windows (R) 10 Enterprise Windows (R) 10 Professional
Windows Server	Microsoft (R) Windows Server (R)
Windows Server 2019	Microsoft (R) Windows Server (R) 2019 Datacenter Microsoft (R) Windows Server (R) 2019 Standard
PAD	Power Automate Desktop Power Automate for desktop
PowerShell	Windows PowerShell

2. 本連携ガイドで説明する連携方法について

JP1 と Power Automate for desktop の連携には、3 種類の方法があります。

(1) デスクトップ連携

Power Automate for Desktop と JP1/CPA が連携し自動化を行う方法です。

3 章で連携方法を説明します。

(2) サーバードesktop連携

Power Automate for Desktop と JP1/CPA が連携、JP1/CPA と JP1/CPA Option for JP1/AJS3、JP1/AJS3 が接続し自動化を行う方法です。

4 章で連携方法を説明します。

(3) Azure 連携

Power Automate for desktop と Power Automate が接続、Power Automate と JP1/AJS3 が連携して自動化を行う方法です。

5 章で連携方法を説明します。

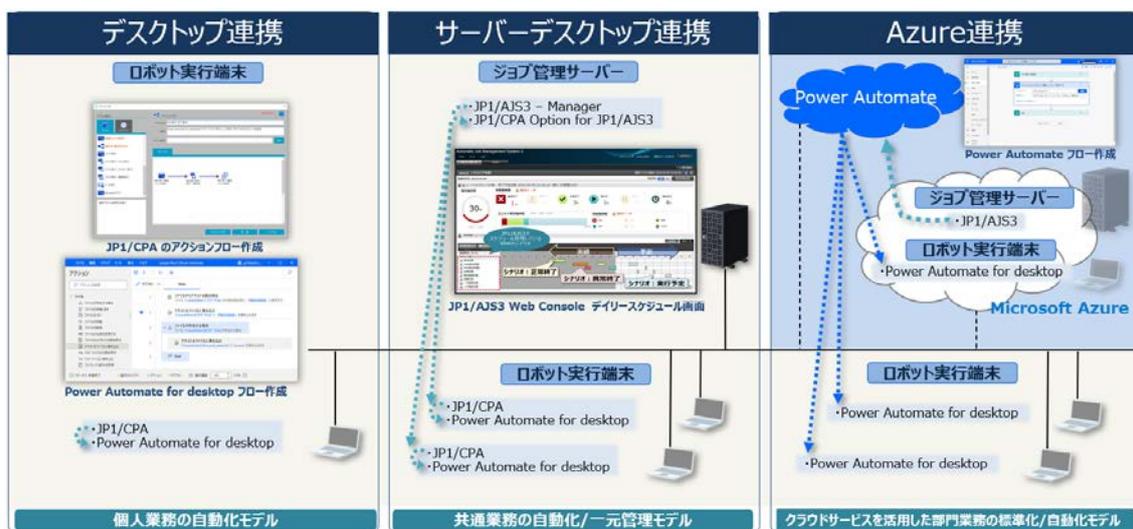


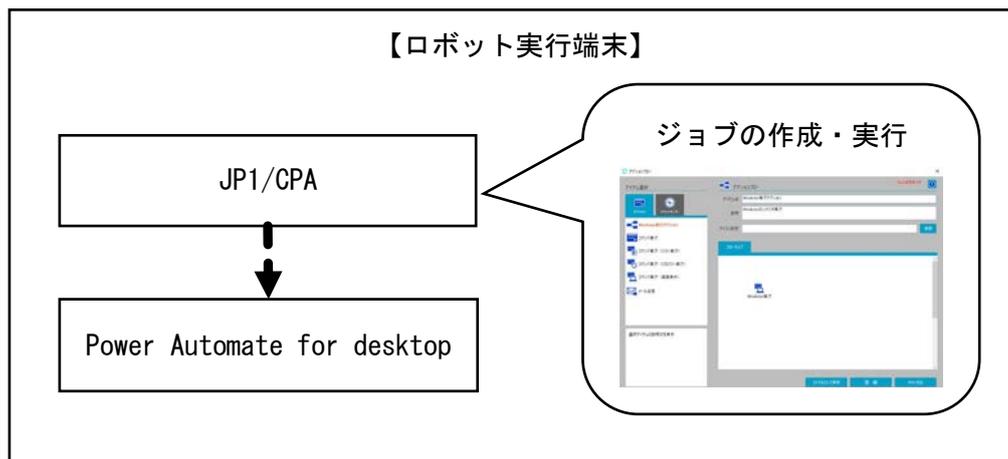
図 2.1 JP1 と Power Automate for desktop の連携モデル

3. デスクトップ連携(個人業務の自動化モデル)

3.1. 概要

3.1.1. JP1/CPA と PAD 連携概要

ロボット端末に存在する JP1/CPA では, Power Automate for desktop の複数のデスクトップフローを JP1/CPA のジョブとして登録することで, 順序制御や定期実行などスケジュール管理することができます。



凡例 :

---▶ : デスクトップフロー実行

図 3.1 概要図

本ガイドでは, ロボット実行端末上の JP1/CPA で Power Automate for desktop のデスクトップフローを実行する方法について説明します。

表 3.1 説明内容

説明内容	説明箇所
Power Automate for desktop のデスクトップフローを JP1/CPA から実行するために必要な Power Automate for desktop の設定	3.2.1 章
Power Automate for desktop のデスクトップフロー実行に必要な JP1/CPA の設定	3.2.2 章
Power Automate for desktop のデスクトップフロー実行	3.2.3 章

3.1.2. 前提プログラム

本ガイドの作成にあたり使用したプログラムおよびバージョンを次に示します。

表 3.2 前提プログラム一覧

対象ホスト	前提プログラム
ロボット実行端末	JP1/CPA 12-50
	Power Automate for desktop 2. 27. 177. 22340 以降

3.1.3. 関連マニュアル

関連するマニュアルを次に示します。

表 3.3 関連マニュアル

製品名	マニュアル名
JP1/CPA	ジョブ管理 基本ガイド（クライアント業務自動化編） JP1/Client Process Automation 構築・運用ガイド

3.1.4. システム構成

システムの構成例を次に示します。

(1) JP1/CPA 連携システム構成

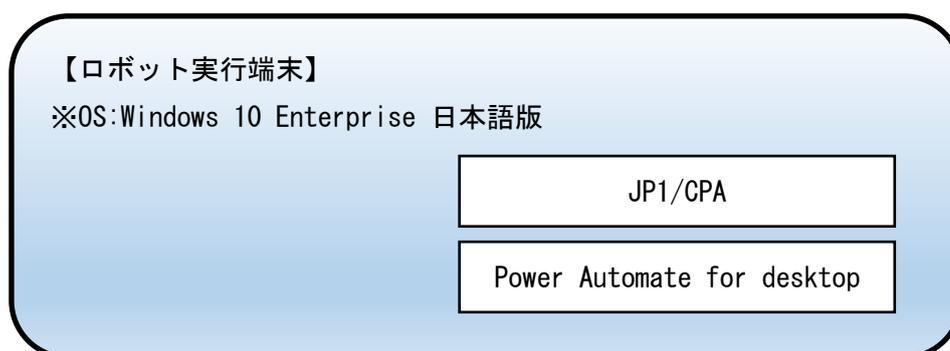


図 3.2 JP1/CPA 連携システム構成例

3.2. JP1/CPA と Power Automate for desktop の連携方法

3.2.1. Power Automate for desktop のデスクトップフローを JP1/CPA から実行するために必要な Power Automate for desktop の設定

Power Automate for desktop のデフォルトでは、コマンドを介したデスクトップフローの実行時は、確認のダイアログが表示され入力待ち状態になるため、確認のダイアログを表示しない設定に変更します。

(1) Power Automate for desktop の設定変更

Power Automate for desktop の設定画面を開き、「フローを外部から呼び出すときに確認のダイアログを表示する」のチェックを外し、「保存」で設定変更を保存します。

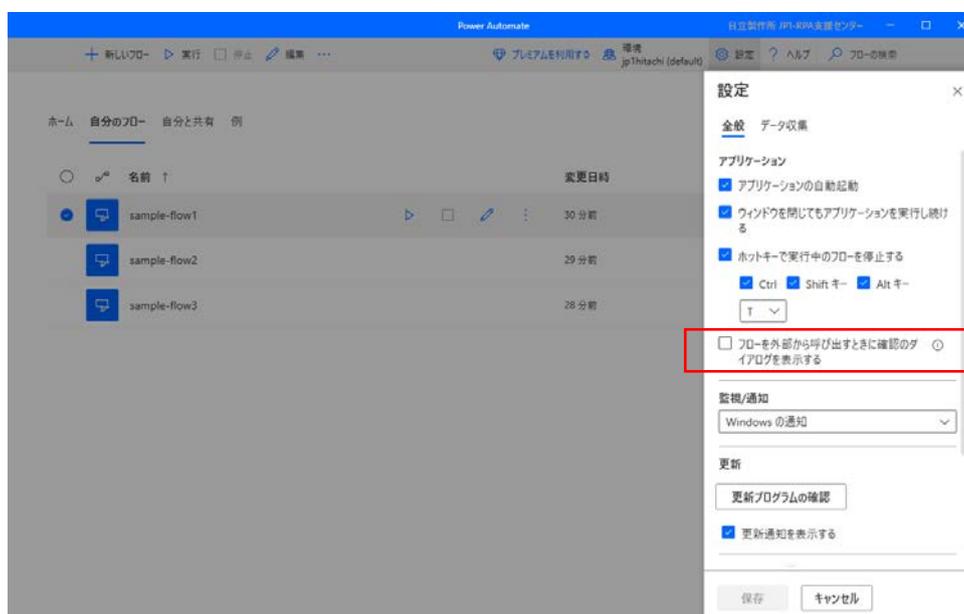


図 3.3 Power Automate for desktop の設定

3.2.2. Power Automate for desktop のデスクトップフロー実行に必要な JP1/CPA の設定

JP1/CPA では、アクションフローに Power Automate for desktop のデスクトップフローを実行するユニットを作成し登録します。JP1/CPA で実施する設定を次に示します。

- (1) JP1/CPA のアクションフローの作成
- (2) Power Automate for desktop 実行ユニットの作成
- (3) JP1/CPA のアクションフローの登録

3.2.2.1. JP1/CPA のアクションフローの作成

Power Automate for desktop のデスクトップフローを実行するアクションフローを作成します。アクションフローの作成方法の詳細については、JP1/CPA のマニュアル「ジョブ管理 基本ガイド (クライアント業務自動化編)」を参照してください。

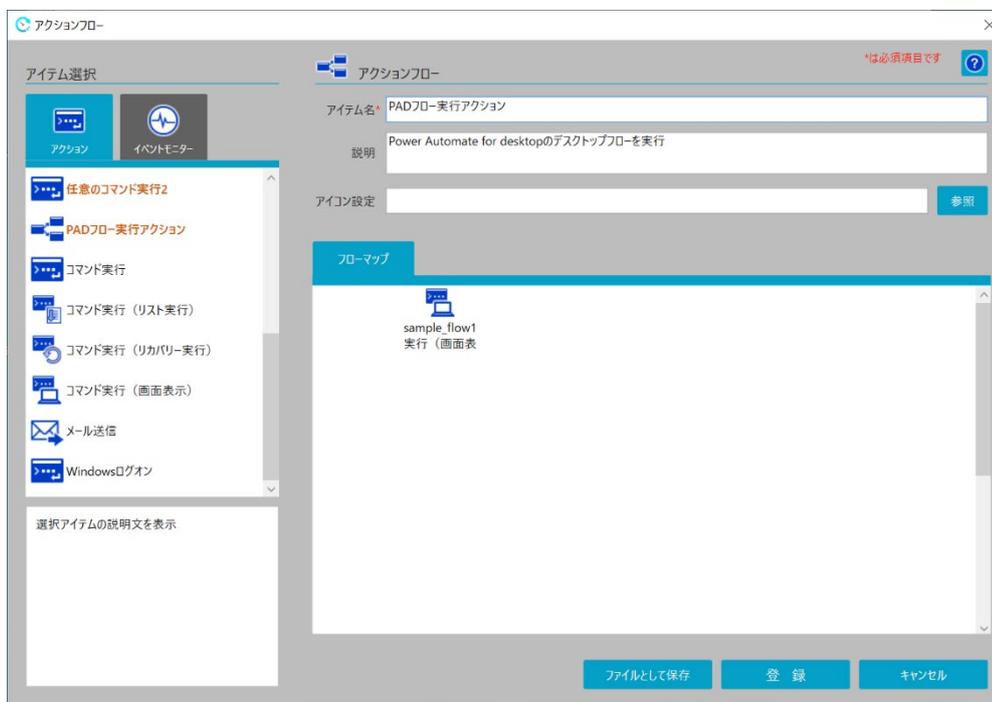


図 3.4 アクションフロー作成例

表 3.4 作成するアクションフローの設定例

設定例	内容
アイテム名	任意の名前です。(PAD フロー実行アクション)
説明	任意のコメントです。

3.2.2.2. Power Automate for desktop 実行ユニットの作成

アクションフローのフローマップに Power Automate for desktop のデスクトップフローを実行するユニットを作成します。Power Automate for desktop 実行ユニットでは Power Automate for desktop のデスクトップフローを起動するための設定が必要になります。

(1) Power Automate for desktop のデスクトップフロー起動方法

Power Automate for desktop のデスクトップフロー起動方法を以下に示します。

● 実行形式

インストールパス¥PAD.Console.Host.exe デスクトップフロー実行 URL

● デスクトップフロー実行 URL

Power Automate for desktop のデスクトップフローの「詳細」を選択し、詳細画面に表示されている「URL を実行」で表示されている URL を、「コピーボタン」にてコピーして取得した URL を指定します。

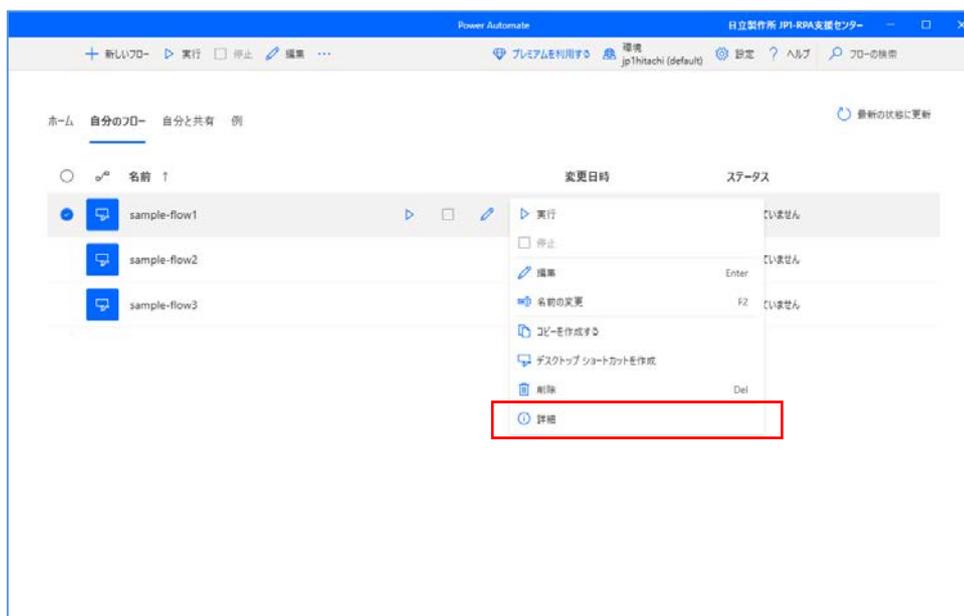


図 3.5 Power Automate for desktop のデスクトップフローの「詳細」の表示

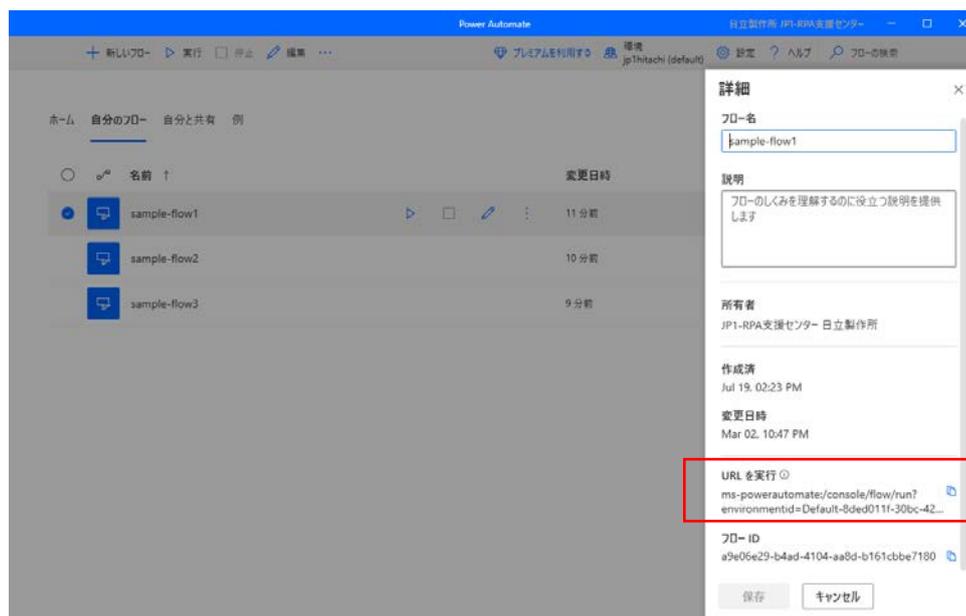


図 3.6 Power Automate for desktop のデスクトップフロー実行 URL の表示

(2) Power Automate for desktop のデスクトップフロー実行ユニット

アクションフロー画面でアクションアイテム「コマンド実行（画面表示）」をフローマップにドラッグ&ドロップし、Power Automate for desktop のデスクトップフローを実行するユニットを作成します。

ユニット	
ユニット名*	sample_flow1実行（画面表示）
アイテム	
アイテム名	コマンド実行（画面表示）
説明	画面を表示するコマンドを実行します。
実行コマンド*	C:¥Program Files (x86)¥Power Automate Desktop¥PAD.C 参照
パラメーター	ms-powerautomate:/console/flow/run?environmentid=Default-8de
終了待ち	画面表示プログラムの終了を待つ
画面終了時の動作	戻り値が0の場合に正常終了
打ち切り時間（分）	3
排他オプション	<input checked="" type="checkbox"/> 指定したアクション間で同時実行しない
OK キャンセル	

図 3.7 Power Automate for desktop のデスクトップフロー実行ユニット作成例

表 3.5 Power Automate for desktop のデスクトップフロー実行ユニットの設定例

設定例	内容
ユニット名	任意の名前です。(sample_flow1 実行 (画面表示))
アイテム名	「コマンド実行 (画面表示)」アイテムを使用します。 ※ : Power Automate for desktop のデスクトップフロー実行は「コマンド実行 (画面表示)」を使用してください。 「コマンド実行」の場合、正常に動作しません。
説明	画面を表示するコマンドを実行します。
実行コマンド	Power Automate for desktop の「PAD.Console.Host.exe」コマンドをフルパスで指定します。
パラメーター	Power Automate for desktop のデスクトップフロー実行 URL を指定します。
終了待ち	「画面表示プログラムの終了を待つ」を選択します。
画面終了時の動作	「戻り値が 0 の場合に正常終了」を選択します。
打ち切り時間	コマンドの実行を打ち切る時間を設定します。 ※ : Power Automate for desktop のデスクトップフロー実行コマンドが予期しないエラーにより終了しない場合、打ち切り時間を設定することで強制終了します。通常、Power Automate for desktop のデスクトップフロー実行コマンドはデスクトップフローが完了するまで待たないため、必要以上に長い時間である必要はありません。
排他オプション	任意に設定します。 ※ : Power Automate for desktop のデスクトップフローは複数のデスクトップフローを同時に実行することができませんが、Power Automate for desktop のデスクトップフローを実行するコマンドはデスクトップフローの実行が完了するまで待たないため、シナリオで複数のデスクトップフローの同時実行をしないよう設計してください。

3.2.2.3. JP1/CPA のアクションフローの登録

アクションフロー画面の「登録」ボタンを押下すると、アクションフローに今回作成したアクションフローが登録されます。

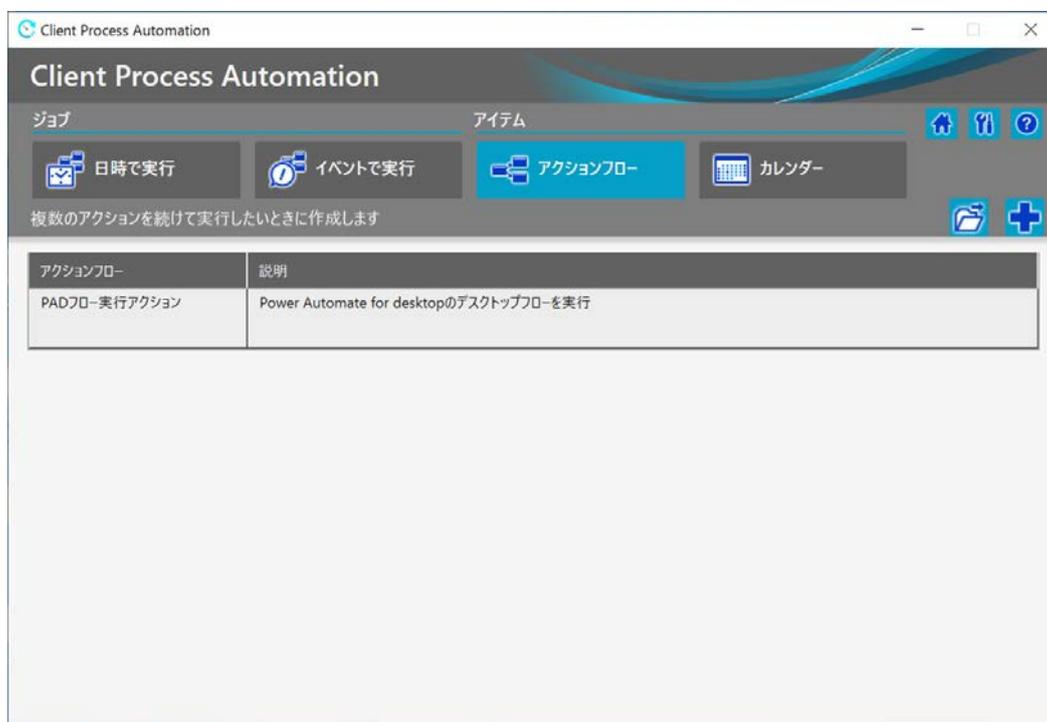


図 3.8 アクションフロー登録例

3.2.3. Power Automate for desktop のデスクトップフロー実行

Power Automate for desktop のデスクトップフローを JP1/CPA のジョブとして実行する方法を次に示します。

- (1) ジョブの作成
- (2) ジョブの実行
- (3) ジョブの実行結果確認

3.2.3.1. ジョブの作成

JP1/CPA では、作成したアクションフローを日時に実行するジョブに設定し登録すると Power Automate for desktop のデスクトップフローを実行します。日時に実行するジョブの作成方法の詳細については、JP1/CPA のマニュアル「ジョブ管理 基本ガイド（クライアント業務自動化編）」を参照してください。

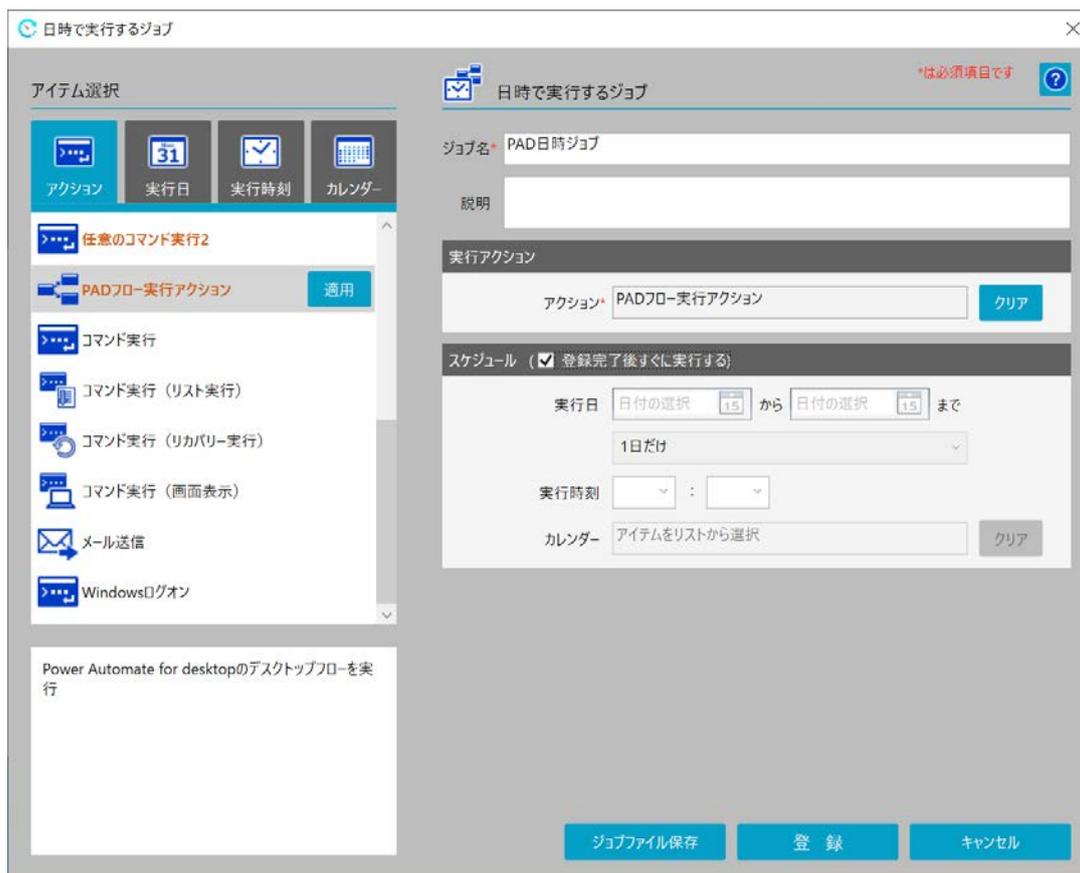


図 3.9 日時に実行するジョブ作成例

表 3.6 日時で実行するジョブの設定例

設定例	内容
ジョブ名	任意の名前です。(PAD 日時ジョブ)
説明	任意のコメントです。
実行アクション	作成したアクションフローをアクションアイテムから選択し適用します。 ※:「3.2.2 Power Automate for desktop のデスクトップフロー 実行に必要な JP1/CPA の設定」で作成したアクションフローを使用します。
スケジュール	「登録完了後すぐに実行する」のチェックボックスをチェックします。 ※: なお、スケジュールは、実行日、実行時刻、カレンダーのアイテムを使用してさまざまな設定が可能です。設定方法の詳細は JP1/CPA のマニュアル「ジョブ管理 基本ガイド (クライアント業務自動化編)」を参照してください。

3.2.3.2. ジョブの実行

日時で実行するジョブの設定画面で「登録」ボタンを押下するとジョブが登録され「実行アクション」に設定したアクションフローが実行されます。

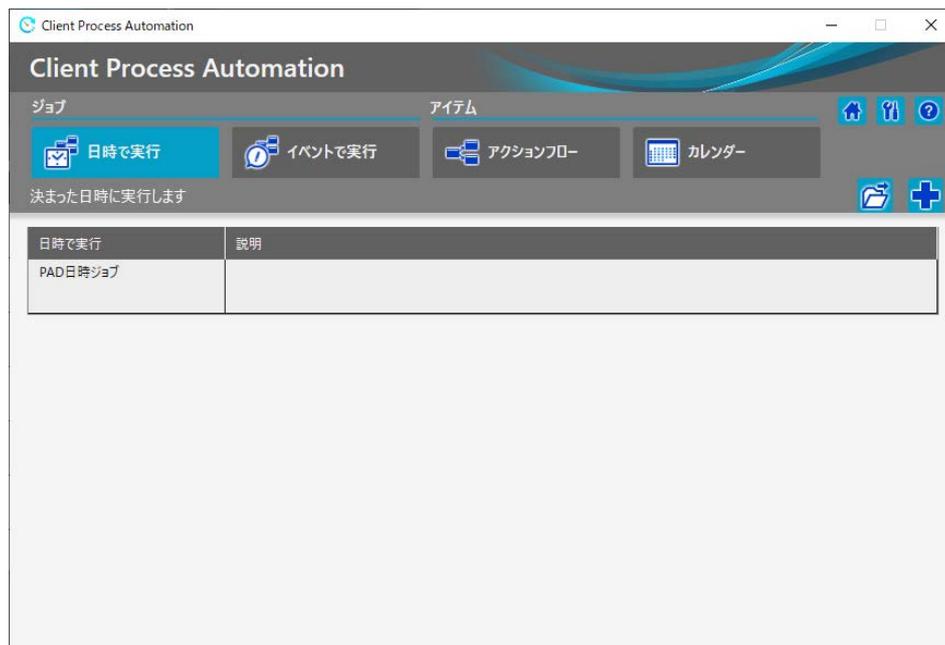


図 3.10 日時で実行するジョブ登録例

3.2.3.3. ジョブの実行結果確認

日時で実行するジョブの一覧，上記で登録したジョブ名を選択し，「状態確認」アイコンをクリックするとジョブの状態が確認できます。

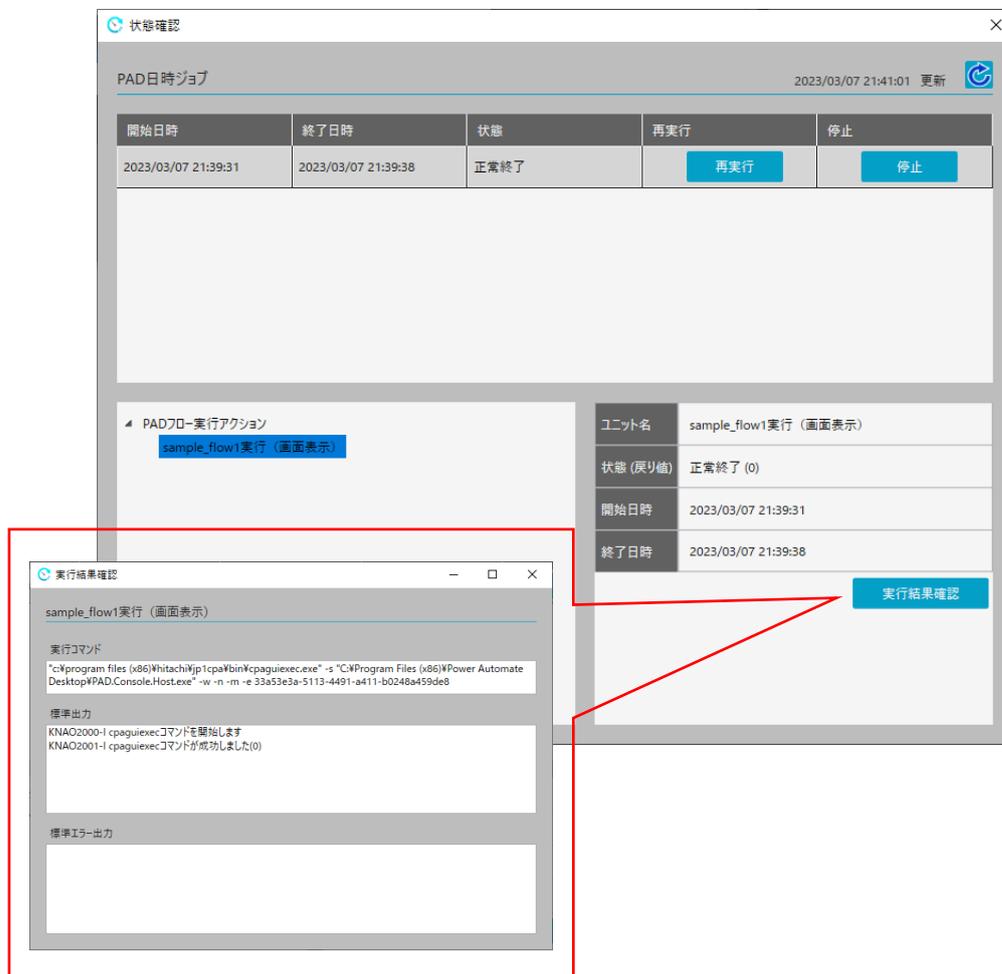


図 3.11 ジョブの実行結果確認例

補足

Power Automate for desktop では，デスクトップフローの実行結果に応じた戻り値を返却する手段がないため，JP1/CPA で実行した Power Automate for desktop のデスクトップフローの実行結果では正常・異常を判断することができません。Power Automate for desktop のデスクトップフローの実行結果の判断方法は Power Automate for desktop のデスクトップフローの動作を考慮し作りこみを行なってください。

3.3. Power Automate for desktop で実行完了ファイルを使用して実行結果を判断するサンプル

「3.2.3.3 Power Automate for desktop のデスクトップフロー実行結果」の補足で説明しましたが、Power Automate for desktop のデスクトップフロー実行結果を確認したい場合は、Power Automate for desktop のデスクトップフローの動作を考慮した作りこみが必要となります。そこで、ここでは、JP1/CPA と Power Automate for desktop で実行完了ファイルを使用して実行結果を判断するサンプルを紹介します。

●サンプルの概要

- ① Power Automate for desktop のデスクトップフロー実行終了確認ファイルを設置するフルパス名を決定する。
- ② Power Automate for desktop のデスクトップフローを作成・変更する。
 - ・ Power Automate for desktop のデスクトップフロー正常終了時に実行終了確認ファイルを作成（テキストをファイルに書き込む）する。
- ③ JP1/CPA のアクションフローを作成する。
 - ・ JP1/CPA で実行するアクションフローを作成する。
 - ・ Power Automate for desktop のデスクトップフロー実行終了確認ファイルを削除する。
 - ・ Power Automate for desktop のデスクトップフローを実行する。
 - ・ イベントモニターのファイル監視で、Power Automate for desktop のデスクトップフロー実行終了確認ファイルを監視する。
 - ・ アクションフローのアイテムに登録する。
- ④ JP1/CPA のジョブに登録する。
- ⑤ JP1/CPA のジョブの実行結果を確認する。

- (1) Power Automate for desktop のデスクトップフロー実行終了確認ファイルを設置するフルパス名を決定

Power Automate for desktop のデスクトップフローの終了を検知するファイルのフルパス名を決定し、ファイルを作成します。(例 : c:\work\check\flow_end_check.txt)

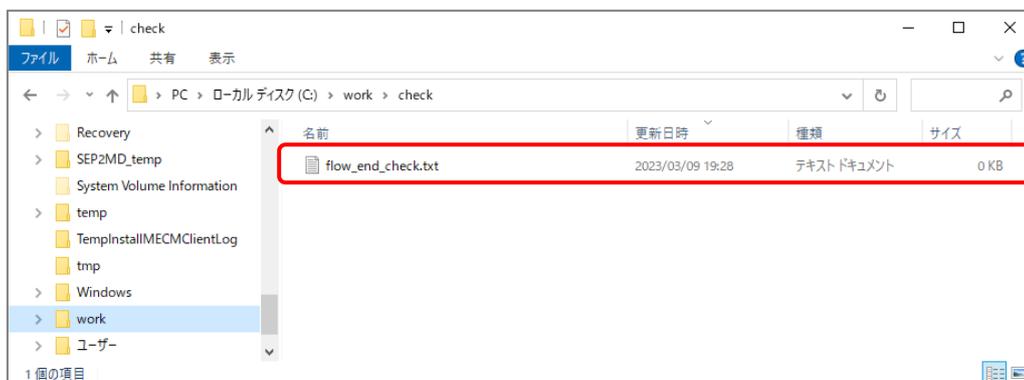


図 3.12 実行ログの設定変更画面

- (2) Power Automate for desktop のデスクトップフローを作成・変更

Power Automate for desktop のデスクトップフローが正常終了する際に、Power Automate for desktop のデスクトップフロー実行終了確認ファイルを作成するようアクションを追加します。(例 : 「テキストをファイルに書き込む」で任意の文字列を書き込むように設定)



図 3.13 アクション追加例

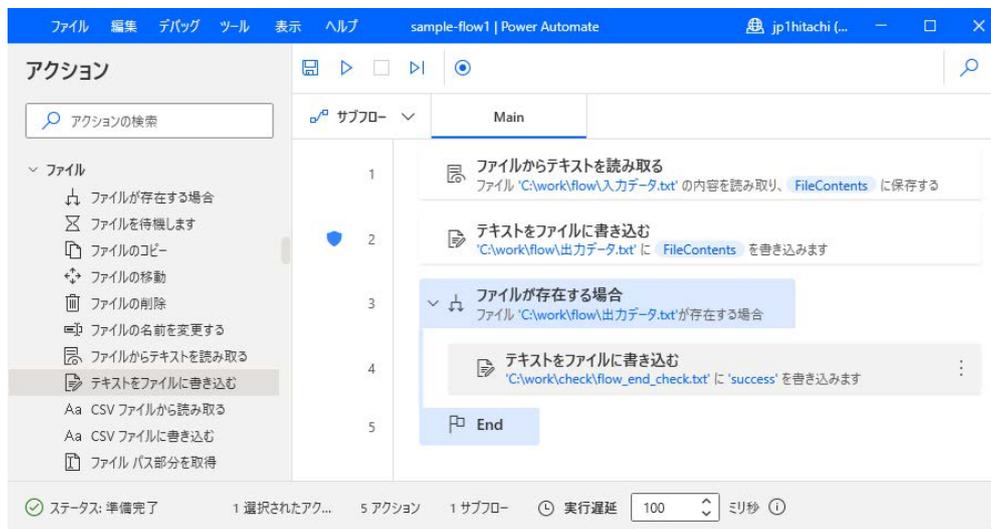


図 3.14 デスクトップフロー作成例

(3) JP1/CPA のアクションフローを作成

(a) アクションフロー作成

JP1/CPA で実行するアクションフローを作成します。

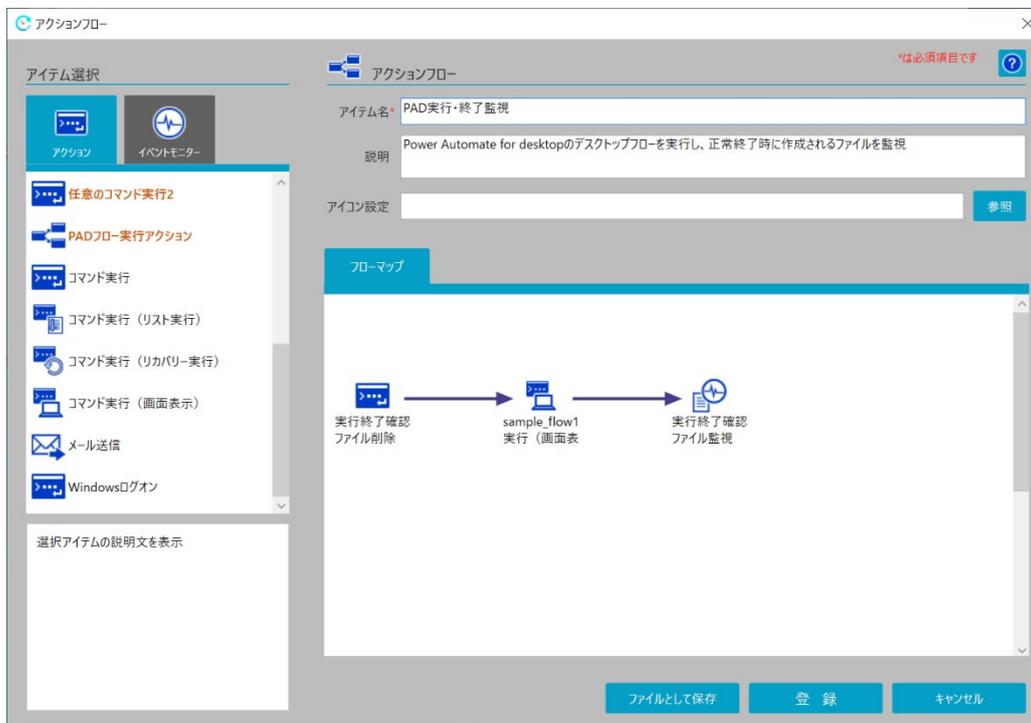


図 3.15 アクションフロー作成例

表 3.7 作成するアクションフローの設定例

設定例	内容
アイテム名	任意の名前です。(PAD 実行・終了監視)
説明	任意のコメントです。

サンプルでは以下の3つのユニットを作成し、「実行終了確認ファイル削除」⇒「PAD コマンド実行（画面表示）」⇒「実行終了確認ファイル監視」の順で実行されるように関連線で接続します。関連線の引き方は、JP1/CPA のマニュアル「ジョブ管理 基本ガイド（クライアント業務自動化編）」を参照して下さい。

表 3.8 フローマップの設定例

ユニット	使用アイテム	動作概要
実行終了確認ファイル削除	コマンド実行アイテム	Power Automate for desktop のデスクトップフローが正常終了時に作成する実行終了確認ファイルを削除します。
sample_flow1 実行（画面表示）	コマンド実行（画面表示）アイテム	Power Automate for desktop のデスクトップフローを実行します。
実行終了確認ファイル監視	「ファイル監視」アイテム	Power Automate for desktop のデスクトップフローが正常終了時に作成する実行終了確認ファイルを監視します。

(b) 実行終了確認ファイル削除ユニット

実行終了確認ファイル削除ユニットでは、Power Automate for desktop のデスクトップフローが正常終了時に作成する実行終了確認ファイルを削除します。

実行終了確認ファイルを削除するユニットの作成例を以下に示します。

●ユニットの設定

図 3.16 実行終了確認ファイル削除ユニット作成例

表 3.9 実行終了確認ファイル削除ユニットの設定例

設定例	内容
ユニット名	任意の名前です。(実行終了確認ファイル削除)
アイテム名	「コマンド実行」アイテムを使用します。
説明	コマンドを実行します。
実行コマンド	C:¥Windows¥System32¥cmd.exe を指定します。 ※OS の環境にあわせて cmd.exe のパスは変更してください
パラメーター	/c del "Power Automate for desktop のデスクトップフロー実行終了確認ファイル"
仮想ユーザー名	特定の Windows ユーザーでコマンドを実行したい場合は、Windows ユーザー名にひも付いた仮想ユーザー名を指定してください。指定しなかった場合は、CPA のジョブ実行サービスのアカウントで実行されます。仮想ユーザー名については、マニュアル「JP1/Client Process Automation 構築・運用ガイド」を参照してください。

(c) PAD コマンド実行（画面表示）ユニット

「3.2.2.2 Power Automate for desktop 実行ユニットの作成」で作成したユニットを使用します。

(d) 実行終了確認ファイル監視ユニット

実行終了確認ファイル監視ユニットでは、Power Automate for desktop のデスクトップフローが正常終了時に作成する実行終了確認ファイルが作成されるのを監視します。

実行終了確認ファイル監視ユニットの作成例を以下に示します。

●ユニットの設定

ユニット

ユニット名* 実行終了確認ファイル監視

アイテム

アイテム名 ファイル監視

説明 ファイルの状態を監視します。

監視対象ファイル名* C:¥work¥check¥flow_end_check.txt 参照

監視条件* 作成

打ち切り時間(分) 2

仮想ユーザー名

OK キャンセル

図 3.17 実行終了確認ファイル監視ユニット作成例

表 3.10 実行終了確認ファイル監視ユニットの設定例

設定例	内容
ユニット名	任意の名前です。(実行終了確認ファイル監視)
アイテム名	「ファイル監視」アイテムを使用します。
説明	ファイルの状態を監視します。

監視対象ファイル名	Power Automate for desktop のデスクトップフローが正常終了時に作成する実行終了確認ファイルを指定します。
監視条件	「作成」を指定します。
打ち切り時間(分)	Power Automate for desktop のデスクトップフローが正常終了時に作成する実行終了確認ファイルが指定時間内に作成されず、デスクトップフローが異常終了したと判断できる時間を指定します。
仮想ユーザー名	入力不要項目です。

(e) アクションフローの登録

アクションフロー画面で「登録」ボタンを押下すると、アクションフローのアイテムに「PAD 実行・終了監視」が登録されます。



図 3.18 アクションフロー登録例

(4) JP1/CPA のジョブ登録

(3) で作成したアクションフローを日時に実行するジョブに設定し登録すると Power Automate for desktop のデスクトップフローを実行します。

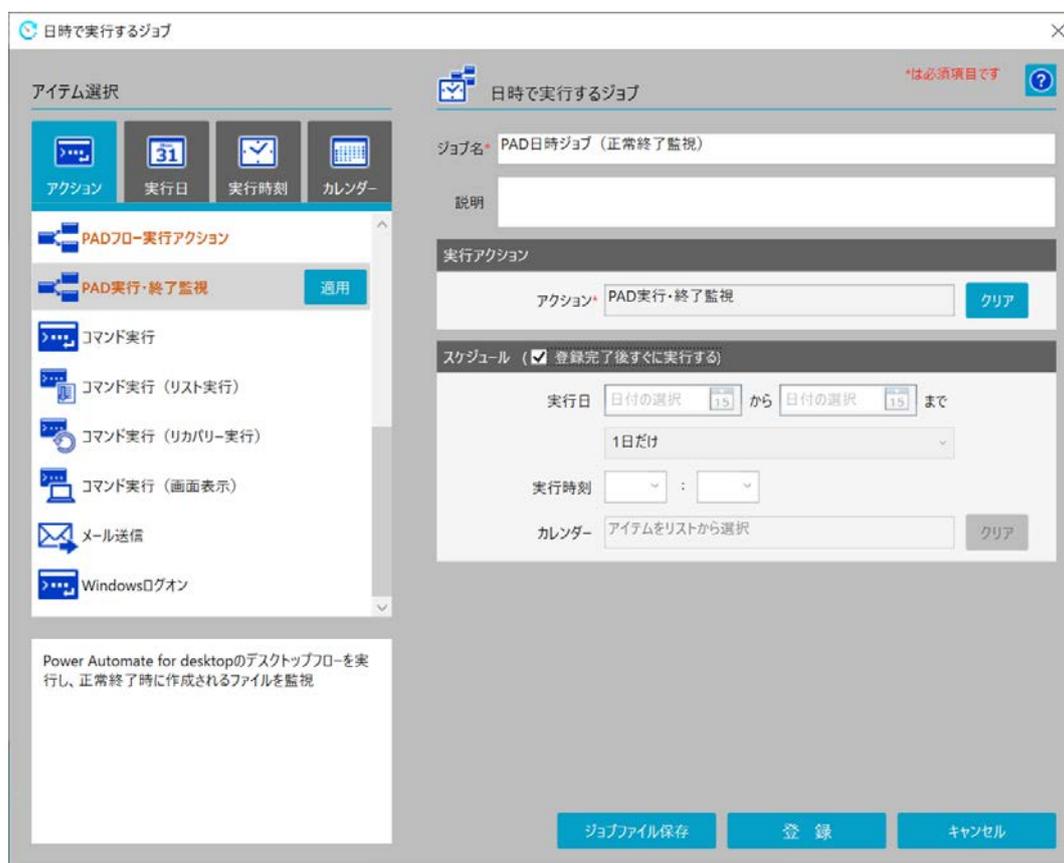


図 3.19 日時に実行するジョブ作成例

表 3.11 日時に実行するジョブの設定例

設定例	内容
ジョブ名	任意の名前です。(PAD 日時ジョブ (正常終了監視))
説明	任意のコメントです。
実行アクション	作成したアクションフローをアクションアイテムから選択し適用します。
スケジュール	「登録完了後すぐに実行する」のチェックボックスをチェックします。

(5) JP1/CPA のジョブの実行結果確認

日時で実行するジョブ「状態確認」に表示されたジョブの結果を選択することで、実行されたアクションフローのユニットごとに結果を確認できます。

The screenshot shows a window titled '状態確認' (Status Confirmation) for a 'PAD日時ジョブ (正常終了監視)' (PAD Time Job (Normal Completion Monitoring)). The window displays a table of job execution results and a detailed view of a specific unit.

開始日時	終了日時	状態	再実行	停止
2023/03/10 18:05:10	2023/03/10 18:06:13	正常終了 ①	再実行	停止
2023/03/10 18:00:40	2023/03/10 18:02:41	異常終了(再実行)	再実行	停止
2023/03/10 17:55:18	2023/03/10 17:57:21	異常終了	再実行	停止

The detailed view shows the following information:

- Unit Name: 実行終了確認ファイル監視
- Status (Refresh): 正常終了 (0)
- Start Date: 2023/03/10 18:05:12
- End Date: 2023/03/10 18:06:13

The detailed view also includes a list of actions under the heading 'PAD実行-終了監視' (PAD Execution - Completion Monitoring):

- 実行終了確認ファイル削除
- sample_flow1実行 (画面表示)
- 実行終了確認ファイル監視 ②

図 3.20 ジョブの実行結果確認例

- ① 「開始日時」や「終了日時」から確認したいジョブの結果を選択します。
- ② 実行したアクションフローを選択することでアクションフローのユニットごとに結果を確認できます。

●異常終了となっているアイテムの確認

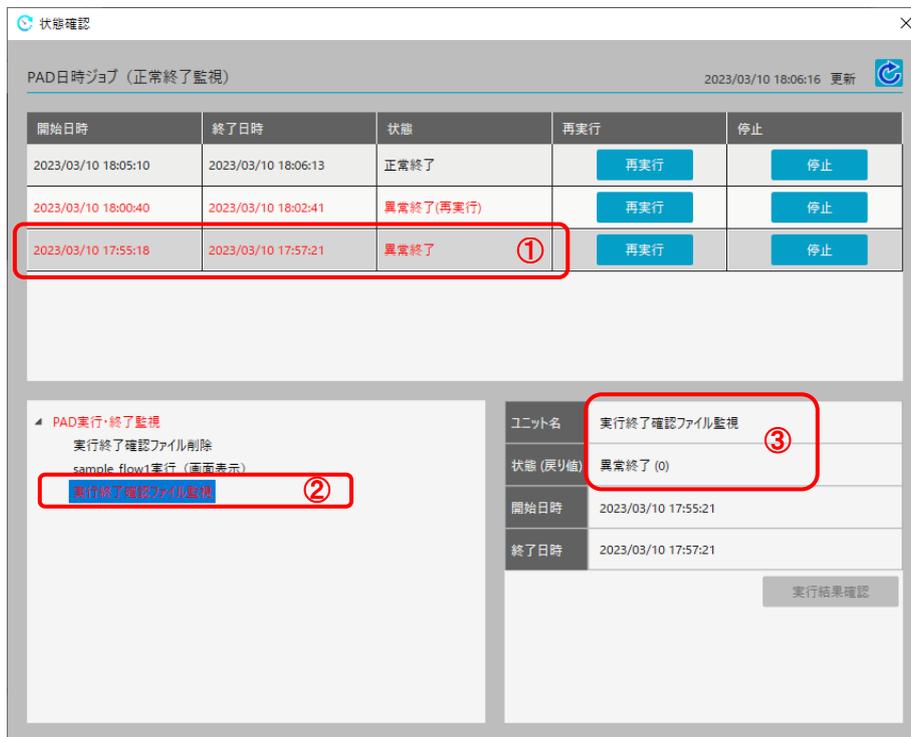


図 3.21 異常終了となっているアイテムの確認例

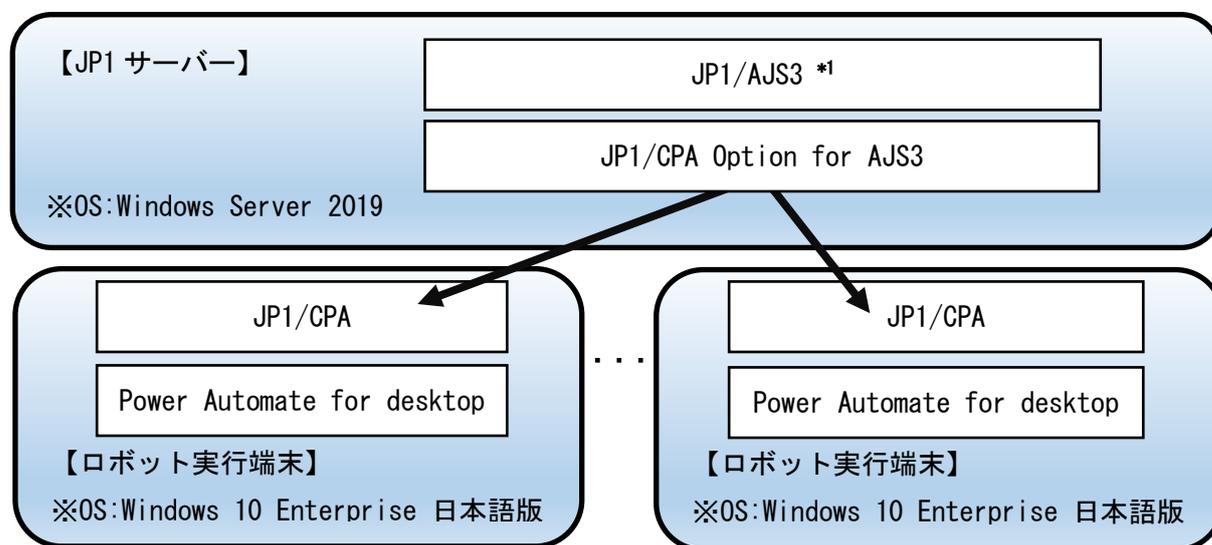
- ① 状態が「異常終了」となっている行を選択することで異常終了したジョブの状態が確認できます。
- ② 実行したアクションフローを選択することで異常終了したユニットを特定することができます。
- ③ 実行終了確認ファイル監視でエラーとなった場合、Power Automate for desktop のデスクトップフローがファイル監視の打ち切り時間内に正常終了せず、タイムアウトとなったことを意味します。

4. サーバーデスクトップ連携(共通業務の自動化/一元管理モデル)

クライアント PC 上の業務を Power Automate for desktop を活用して自動化する際、実行する業務のスケジュールリングや、業務の実行結果の確認等を JP1/CPA を用いることで、効率的に実行することができます。

このようなクライアント PC 上の業務自動化の適用範囲が拡大し、対象クライアント PC の台数が増加した場合、個々のクライアント PC ごとに行っていた JP1/CPA での運用管理を、JP1/AJS3 での一元管理に移行することで、より効率的に適用範囲を拡大することができます。

(1) JP1/AJS3+JP1/CPA 連携システム構成



*1 : JP1/AJS3 Version 12 以前の場合は、JP1/Baseが必要です。

図 4-1 JP1/AJS3+JP1/CPA 連携システム構成例

JP1/AJS3 と連携することで JP1/CPA 上で登録済みのジョブを、JP1/AJS3 のジョブネット内に定義する JP1/CPA 連携用カスタムジョブから起動できます。

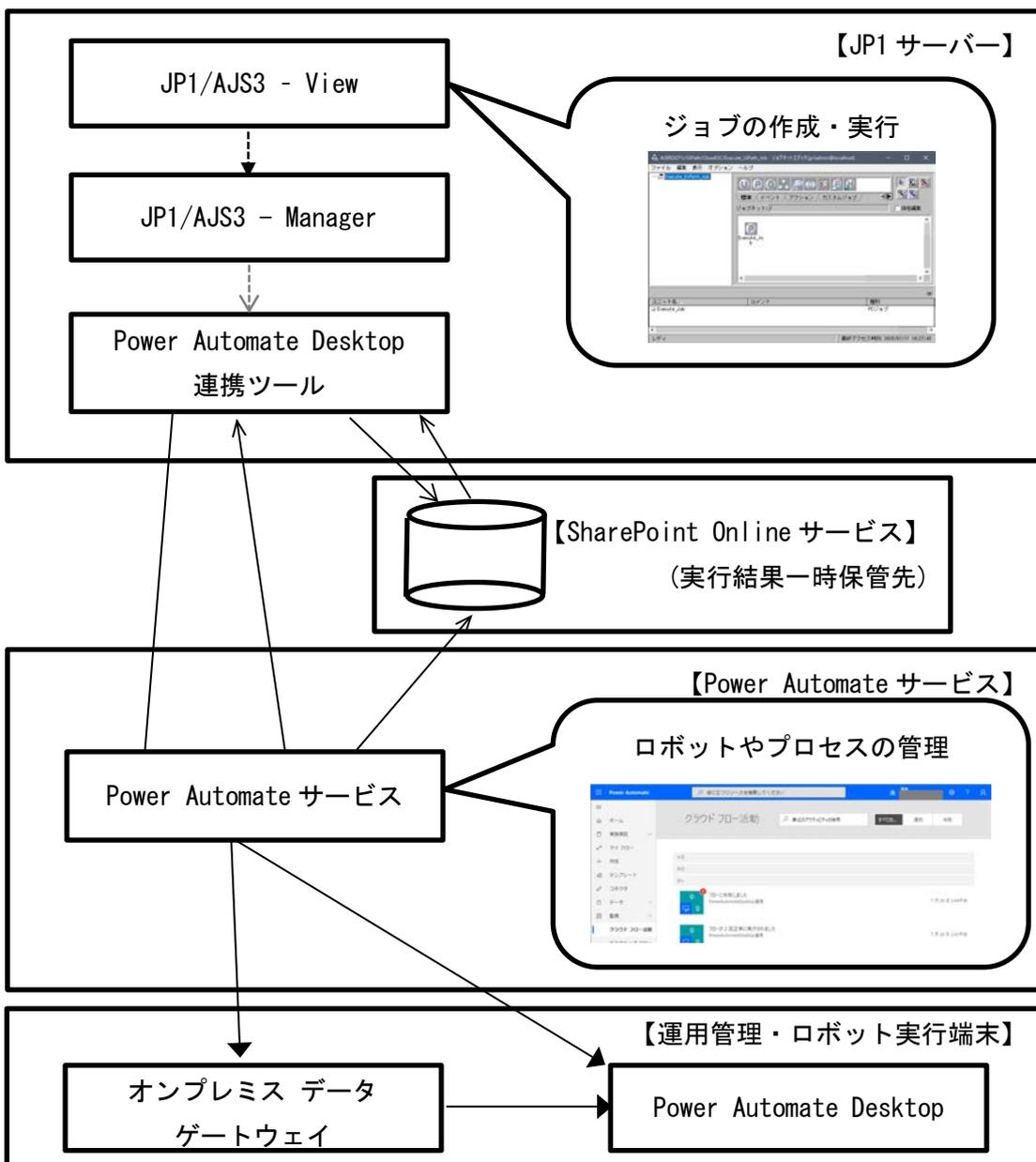
JP1/AJS3 連携機能の詳細については、JP1/CPA のマニュアル「JP1/Client Process Automation 構築・運用ガイド」を参照してください。

5. Azure 連携(クラウド部門業務の標準化/自動化モデル)

5.1. 概要

5.1.1. Power Automate for desktop 連携ツール 概要

JP1/AJS3 のジョブで Power Automate for desktop 連携ツールを起動し、ロボットを実行することで、ロボット実行端末に定義されているプロセスの順序制御や定期実行など実行スケジュールを管理することができます。Power Automate for desktop 連携ツールの提供に関しては、JP1 テクニカルサービス窓口までお問い合わせください。



【凡例】

----> : ジョブの作成・実行

----> : Power Automate for desktop 連携ツールの実行

—> : プロセスの実行依頼または応答

—> : プロセスの実行

5.1.2. Power Automate for desktop 連携ツール メリットおよび注意事項

PowerAutomate for desktop 連携ツールを利用することのメリットを以下に示します。

表 5.1 Power Automate for desktop 連携ツールを利用した場合のメリット

#	メリット	説明
1	JP1 で全ロボットが管理可能	JP1 からロボットのスケジュール管理（予実管理）が可能となります。
2	RPA 実行端末で動作するロボットのフローも JP1 の 1 つのジョブとして登録し、ジョブネットとして実施・管理可能	JP1 のジョブスケジューラーから RPA も管理できるようになります。
3	ロボットの実行結果が JP1 上で管理可能	ロボットの動作不良が起きた場合の状況把握も JP1 上で容易に確認が可能となります。

5.1.3. JP1/AJS3 と Power Automate for desktop との連携に関する前提条件

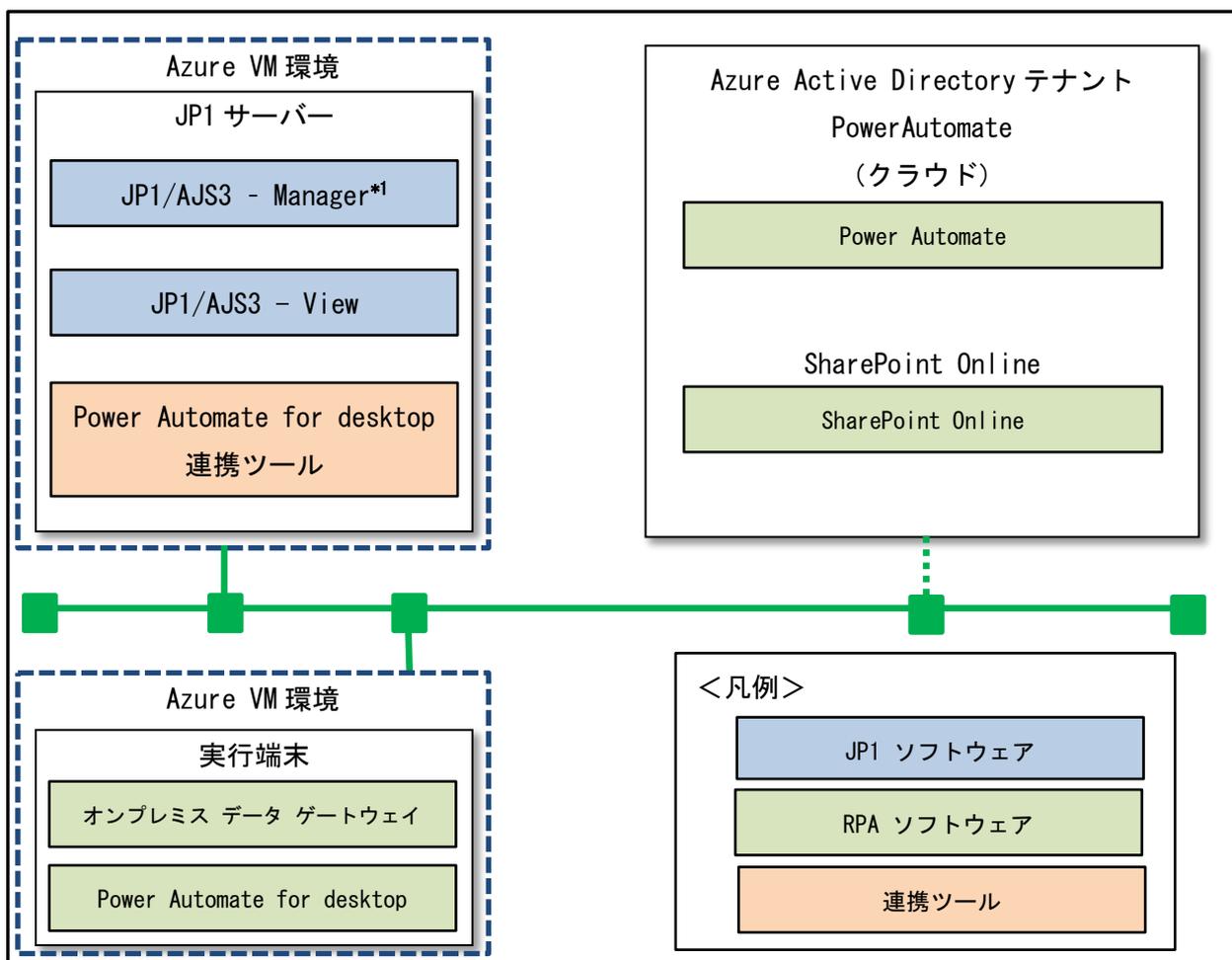
- (a) JP1 サーバー（JP1/AJS3 - Manager が導入されている端末）と、Power Automate サービス（クラウドサービス）の通信が可能であること。
- (b) Power Automate サービス（クラウドサービス）と、ロボット実行端末（Power Automate for desktop、オンプレミス データ ゲートウェイ※1 が導入されている端末）の通信が可能であること。
- (c) Power Automate for desktop と Power Automate サービス（クラウドサービス）の接続を行うため、以下のサービス、アカウントの利用が可能であること。
 - ・ Microsoft365 テナント（Power Automate のクラウドフローが実行できる環境）
 - ・ 上記テナント内のユーザー（開発・実行を行う）
 - ・ Azure AD の管理者（デバイスの登録の確認を行う）※1
- (d) Power Automate サービス（クラウドサービス）および JP1 サーバー（JP1/AJS3 - Manager が導入されている端末）から、SharePoint Online チームサイトに対して以下の操作が可能であること。
 - ・ Power Automate のクラウドフローから、SharePoint Online チームサイトへのファイル作成

- ・ JP1 サーバー（JP1/AJS3 - Manager が導入されている端末）から SharePoint Online チームサイトへのファイル参照・削除
- (e) ロボット実行端末（Power Automate for desktop が導入されている端末）に Power Automate サービス（クラウドサービス）から直接接続して（オンプレミス データ ゲートウェイを利用せず）Desktop フローを実行する場合は、Power Automate サービスにロボット実行端末が登録されていること。
- ※1：オンプレミス データ ゲートウェイを利用する場合

Power Automate for desktop 連携ツール 導入構成

Power Automate for desktop 連携ツールの導入構成については、以下の通りです。

※本ガイドは、Azure VM 環境下での構成例となります。



*1：JP1/AJS3 Version 12 以前の場合は、JP1/Base が必要です。

5.2. Power Automate for desktop と JP1/AJS3 の連携方法

5.2.1. 前提条件

Power Automate for desktop 連携ツールの前提となる OS およびソフトウェアを以下に示します。

表 5.2 前提 OS 一覧

製品名	バージョン
Windows	10 Professional/Enterprise 以降
Windows Server	2019 以降

表 5.3 前提ソフトウェア一覧

製品名	バージョン	
Power Automate for desktop	Power Automate for desktop 2.7.49.21099 以降	
JP1	JP1/AJS3 - Manager	11-00 以降
	JP1/AJS3 - Agent	11-00 以降
	JP1/AJS3 - View	11-00 以降
	JP1/Base	11-00 以降、12-50 以前
PowerShell	5 以降	
SharePoint Online Client Components SDK	16.0.6906.1200 以降	

※SharePoint Online Client Components SDK については、次のリンクから環境に応じたファイルを、JP1/AJS3 - Manager 環境へダウンロード/インストールしてください。

<https://www.microsoft.com/en-us/download/details.aspx?id=42038>

5.2.2. Power Automate for desktop 環境設定手順

オンプレミス データ ゲートウェイを利用する場合に必要な設定手順となります。
オンプレミス データ ゲートウェイを利用しない場合は、「5.2.3 Power Automate クラウドフロー作成手順」へお進みください。

5.2.2.1. Power Automate for desktop 環境での、Microsoft アカウント設定方法

Power Automate for desktop は、Power Automate から管理する必要があります。この場合 Azure Active Directory のテナントにサインイン可能な環境で実行する必要があるため、本ガイドでは Microsoft Azure 環境上の Windows10 日本語版にて、Azure Active Directory へデバイス登録を行うための手順を以下に示します。

本ガイドは、Azure VM 環境下での設定手順例となります。

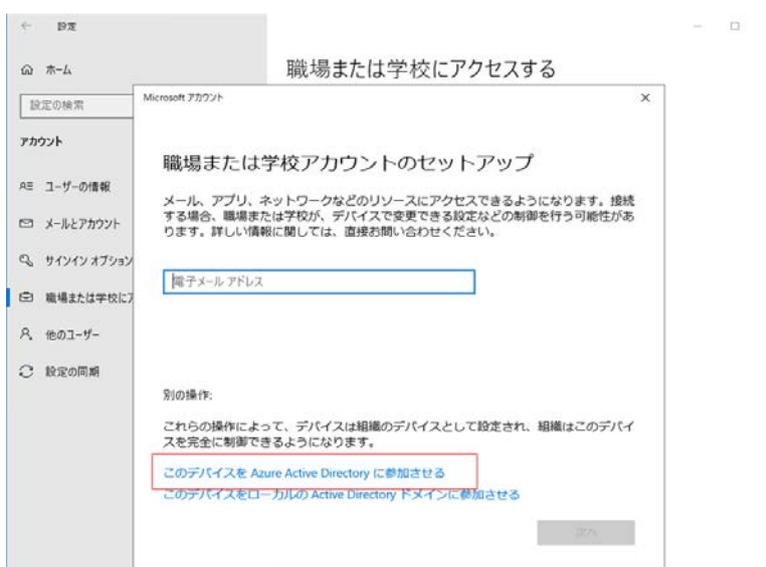
- (1) Azure Active Directory へのデバイス登録をします。
 - (a) Administrator 権限を持つユーザーで Windows10 日本語版にログインします。
 - (b) 次に Windows10 日本語版の「設定」⇒「アカウント」を開き、「ユーザーの情報」に「Microsoft アカウントでのサインインに切り替える」が無いことを確認し、「職場または学校にアクセスする」を選択します。



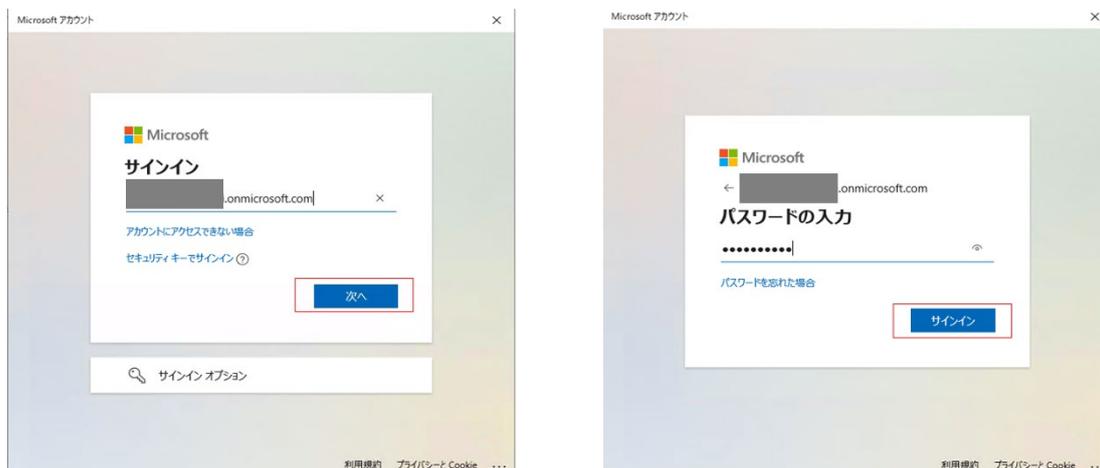
(c) 「職場または学校にアクセスする」から、接続を選択します。



(d) 「職場または学校アカウントのセットアップ」から、「このデバイスを Azure Active Directoryに参加させる」を選択します。



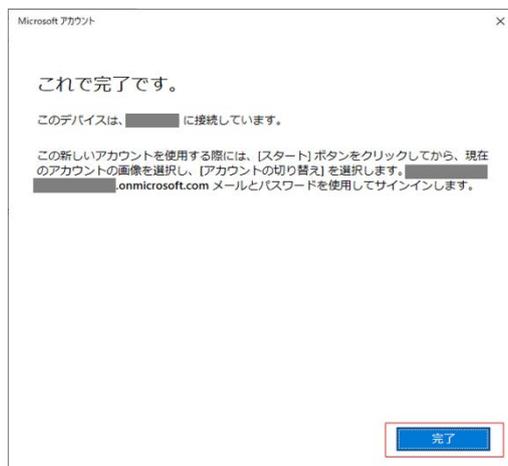
- (e) 「Azure AD テナント」へのサインイン画面が開きますので、Power Automateで使用する Azure AD テナント内のアカウントを入力し「次へ」を選択するとパスワード入力画面に遷移しますので、パスワードを入力し「サインイン」を選択します。



- (f) 「これがあなたの組織のネットワークであることを確認してください」画面が開きますので、「接続先」「ユーザー名」に誤りが無いことを確認し、「参加する」を選択します。



- (g) 登録結果の画面に「これで完了です」が表示されたら、「完了」を選択します。

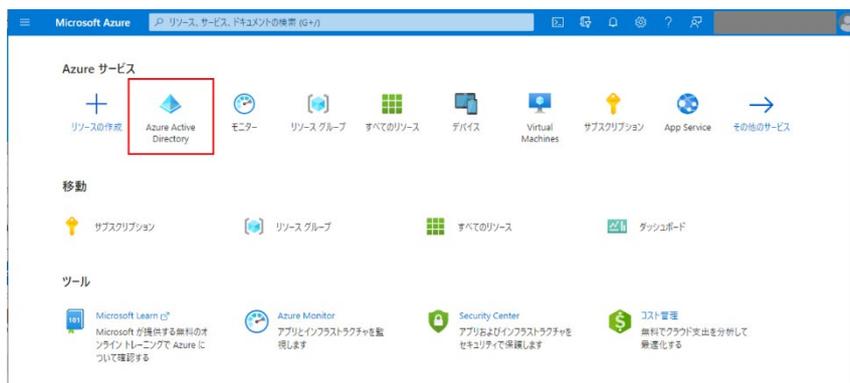


- (h) 「職場または学校にアクセスする」画面に戻り、「Azure ADに接続済み」と表示されていることを確認します。



- (2) Azure Active Directory での確認をします。

- (a) <https://portal.azure.com/#home> へブラウザからアクセスし、Azure AD の管理者アカウントでサインインし、「Azure Active Directory」を選択します。



- (b) 「Azure Active Directory」画面から、「デバイス」を選択します。

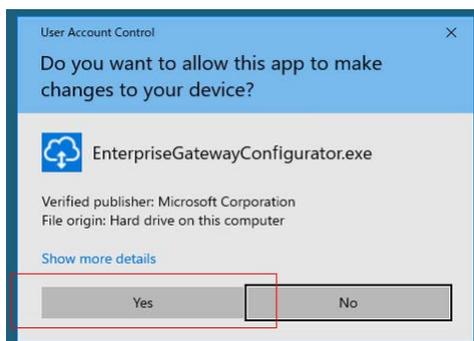


- (c) 「デバイス | すべてのデバイス」画面に、今回 Microsoft アカウントに参加した Azure VM のデバイス名が表示されていることを確認します。

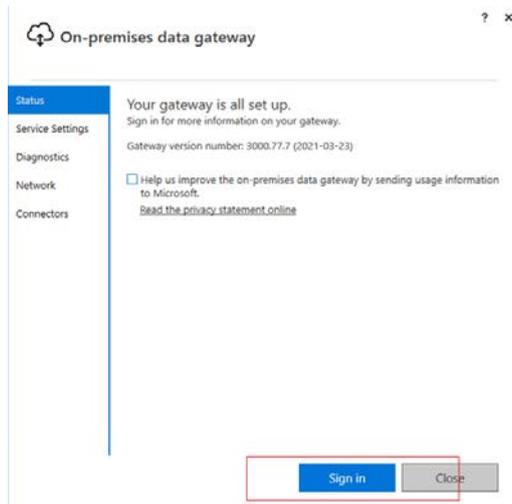


- (3) Windows10 日本語版環境での、オンプレミス データ ゲートウェイの設定をします。

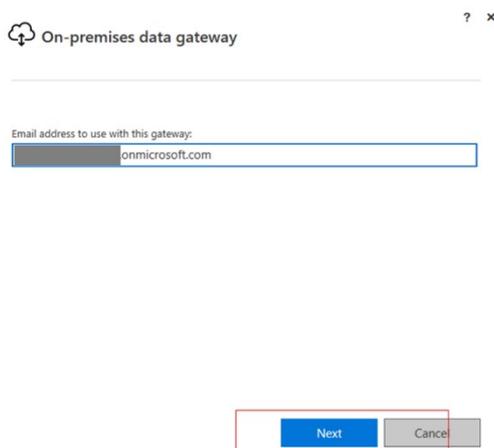
- (a) Administrator 権限を持つユーザーで Windows10 日本語版にログインします。
- (b) Windows10 日本語版の「On-premises data gateway」を開き、UAC ダイアログが出た場合は、「Yes」を選択します。



- (c) 「On-premises data gateway」画面の「Status」を選択し、「Sign in」を選択します。

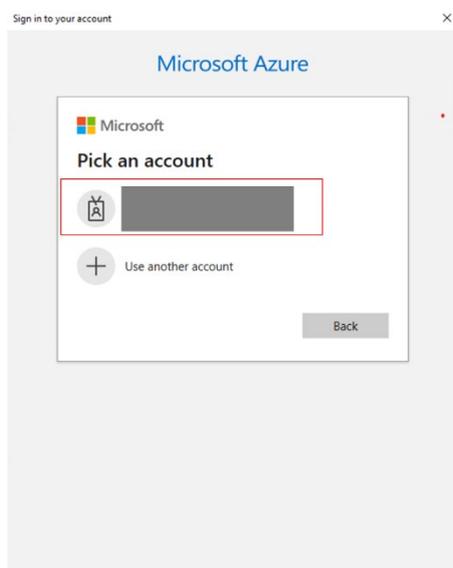


- (d) 「Email address to use this gateway」の個所に、Power Automateを開発するユーザーのメールアドレスを入力し、「Next」を選択します。Power Automateを開発するユーザーはPowerAutomateが動作しているAzure Active Directoryテナントのユーザーである必要があります。

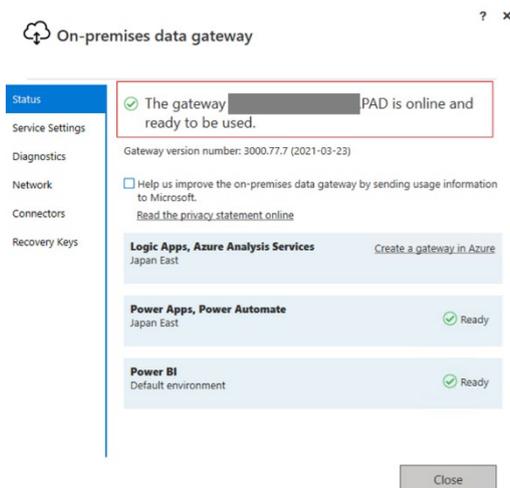


(e) 「Sign in to your account」画面が開きます。対象の Power Automate を開発するユーザーを選択します。

※パスワードの入力を求められる場合があります。この場合は、Power Automate を開発するユーザーのパスワードを入力してください。



(f) 「On-premises data gateway」画面の「Status」に戻ります。赤枠の「The Gateway」の後に続く名称をメモしておきます（(5) Power Automate で接続の確認をします。にて接続確認を行う際に必要となります）。その後「Close」を選択します。



(4) AZURE 上の Win10 のシステムロケール（コントロールパネルの地域）の設定をします。

Power Automate のサインインアカウントにされている地域設定と、Windows10 日本語版環境のシステムロケールを同一に設定します。

- (a) <https://japan.flow.microsoft.com/ja-jp/> へブラウザからアクセスし、Power Automate を開発するユーザーでサインインし、赤枠で囲んだ個所を選択の後「アカウントを表示」を選択します。

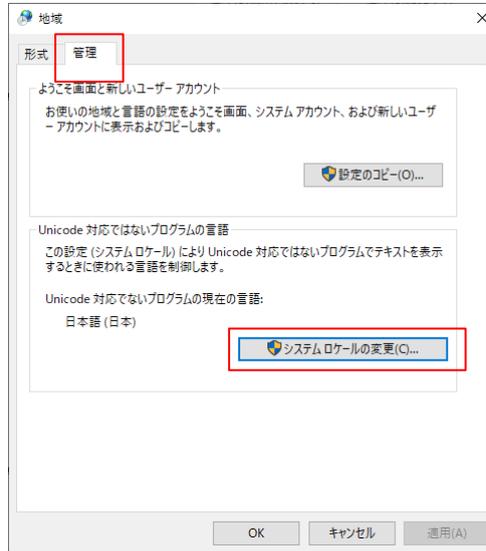


- (b) 「マイアカウント」画面が新たに開きます。「設定およびプライバシー」を選択し、地域設定を確認します。

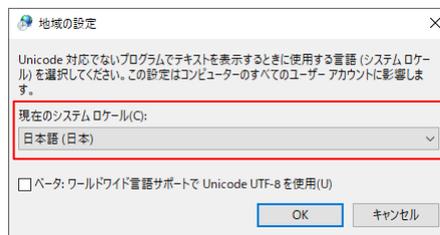


- (c) Administrator 権限を持つユーザーで Windows10 日本語版にログインします。

- (d) Windows10 日本語版のコントロールパネルから「地域」を開き、「管理」の「システムロケールの変更」を選択します。



- (e) 「地域の設定」画面が開きます。「現在のシステムロケール」が Azure AD アカウントの「地域設定」と異なる場合は、同一の地域設定となるよう変更します。



(5) Power Automate で接続の確認をします。

- (a) <https://japan.flow.microsoft.com/ja-jp/> へブラウザからアクセスし、Azure AD アカウントでサインインし、「データ」⇒「ゲートウェイ」を選択します。オンプレミス データ ゲートウェイの設定時にメモしたゲートウェイの名称を選択します。



- (b) ゲートウェイの詳細画面へ遷移します。状態が「オンライン」の状態であることを確認します。



5.2.3. Power Automate クラウドフロー作成手順

5.2.3.1. Power Automate for desktop を実行するための、Power Automate クラウドフロー作成手順

JP1/AJS3 から Power Automate for desktop のデスクトップフローを実行するために、Power Automate のクラウドフローを作成し、クラウドフローから Power Automate for desktop のデスクトップフローを実行する必要があります。

このため Power Automate のクラウドフローを作成する手順を以下に示します。

(1) Power Automate にクラウドフローを作成します。

- (a) <https://japan.flow.microsoft.com/ja-jp/> へブラウザからアクセスし、Power Automate を開発するユーザーでサインインし、「マイフロー」を選択します。



- (b) 「マイフロー」画面から「+新しいフロー」を選択し、「インスタント クラウドフロー」を選択します。



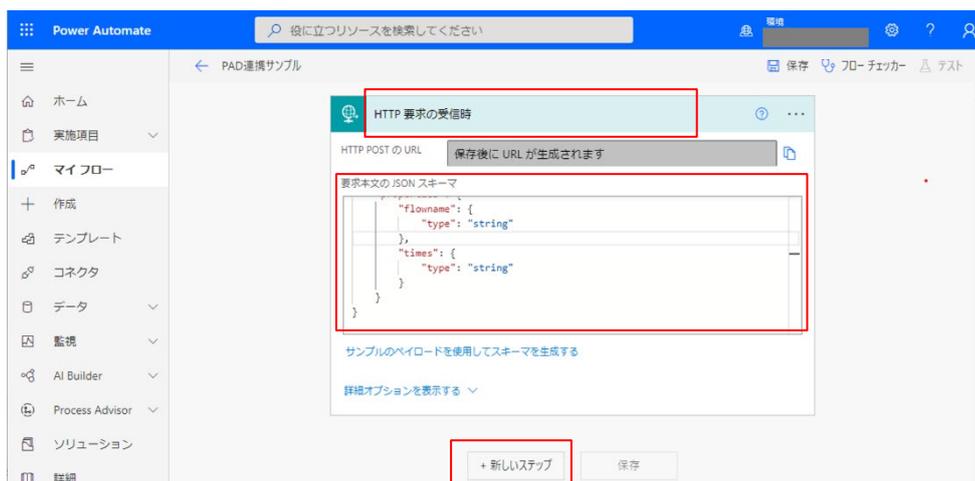
- (c) 「インスタント クラウドフローを構築する」画面から「フロー名」に任意のクラウドフロー名称を入力します。次に、「このフローをトリガーとする方法を

選択します。」から、「HTTP 要求の受信時」を選択し、「作成」を選択します。

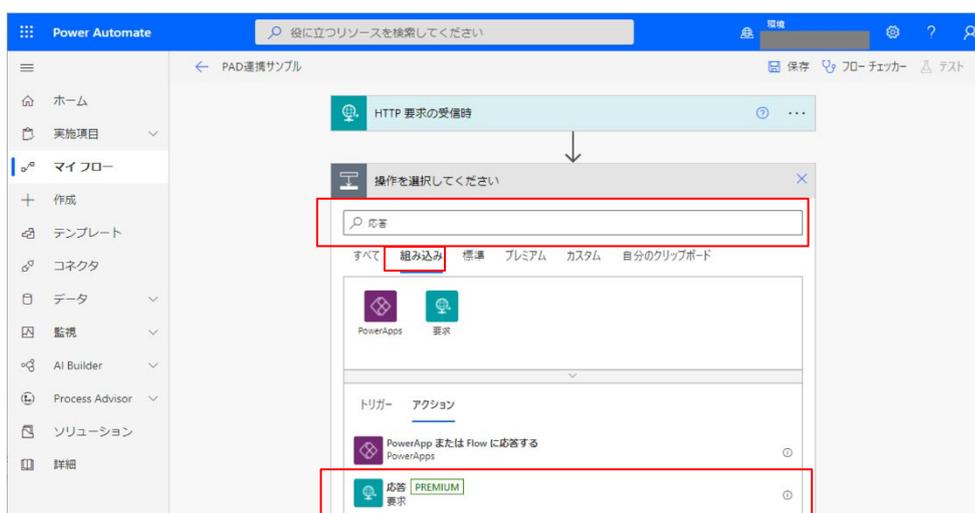


(d) 「フローの作成」画面に遷移します。「HTTP 要求の受信時」を選択し、「要求本文の JSON スキーマ」に次の JSON を入力し、「+新しいステップ」を選択します。

```
{
  "type": "object",
  "properties": {
    "flowname": {
      "type": "string"
    },
    "times": {
      "type": "string"
    }
  }
}
```

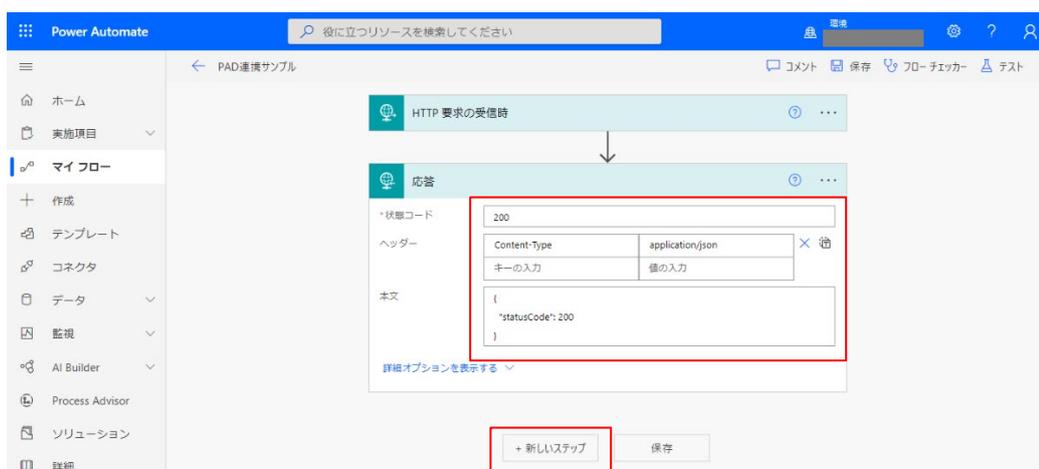


- (e) 新しいステップとして「操作を選択してください」がフローに追加されます。検索窓に「応答」と入力し、「組み込み」を選択します。次に「アクション」の「応答」を選択します。



- (f) 「応答」の設定が開きます。「状態コード」に「200」を入力します。「ヘッダー」の「キーの入力」個所に「Content-Type」、「値の入力」個所に「application/json」を入力します。次に「本文」の箇所へ以下の応答内容を入力し、「+新しいステップ」を選択します。

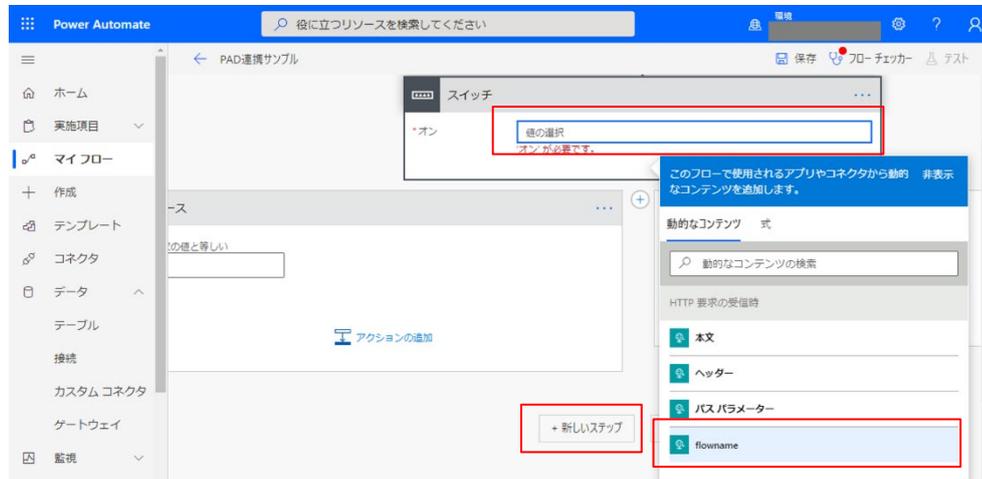
```
{
  "statusCode": 200
}
```



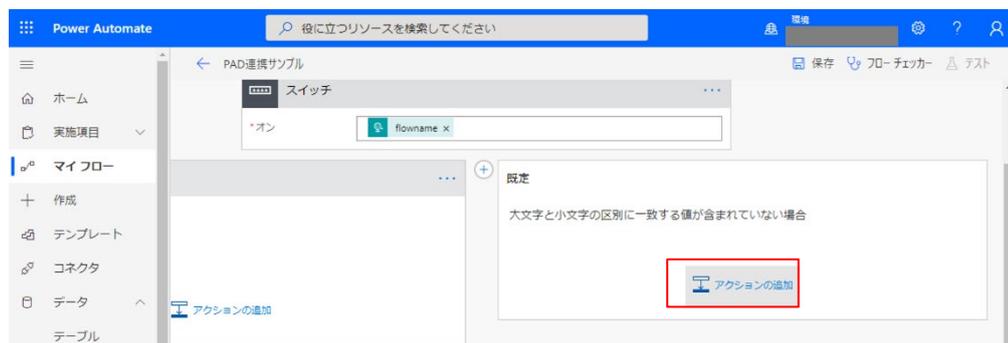
- (g) 新しいステップとして「操作を選択してください」がフローに追加されます。検索窓に「スイッチ」と入力し、「アクション」の「スイッチ」を選択します。



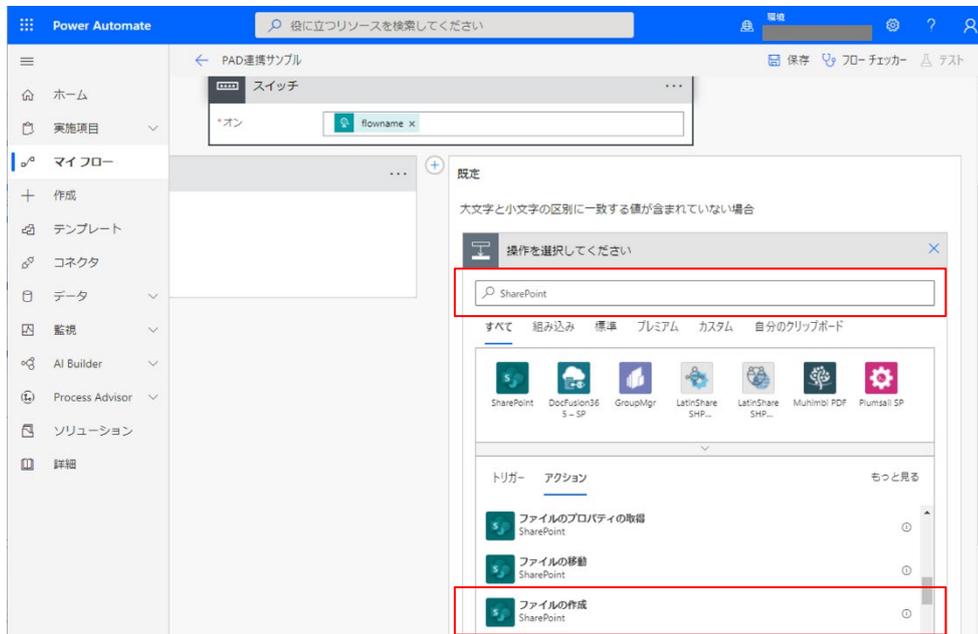
- (h) 「スイッチ」の設定が開きます。「値の選択」を選択し、「動的なコンテンツ」から「flowname」を選択します。



- (i) 「スイッチ」で分岐された「既定」にある「アクションの追加」を選択します。



- (j) 新しいステップとして「操作を選択してください」がフローに追加されます。検索窓に「SharePoint」と入力し、「ファイルの作成」を選択します。



- (k) 「ファイルの作成」の設定が開きます。「サイトのアドレス」に「カスタム値の入力」を選択し、アクセス可能な SharePoint Online チームサイトの URL を入力します（この URL については Power Automate for desktop 連携ツールでの、 $\$SpoSiteURL$ パラメーター設定値になります）。

次に「フォルダーのパス」の右端にある「ピッカーの表示」を選択し、対象の SharePoint Online チームサイトにある、データ連携用フォルダを選択してください（本ガイドでは、SharePoint Online チームサイトの /Shared Documents/PAD_Log_Temp 配下をデータ連携用フォルダの例としており、このフォルダについては Power Automate for desktop 連携ツールでの、 $\$SpoStaticURL$ パラメーター設定値になります）。

次に「ファイル名」の個所を選択し、

「`@{triggerBody()?['flowname']}_@{triggerBody()?['times']}.json`」を入力すると、画面のようなファイル名の表示になります。

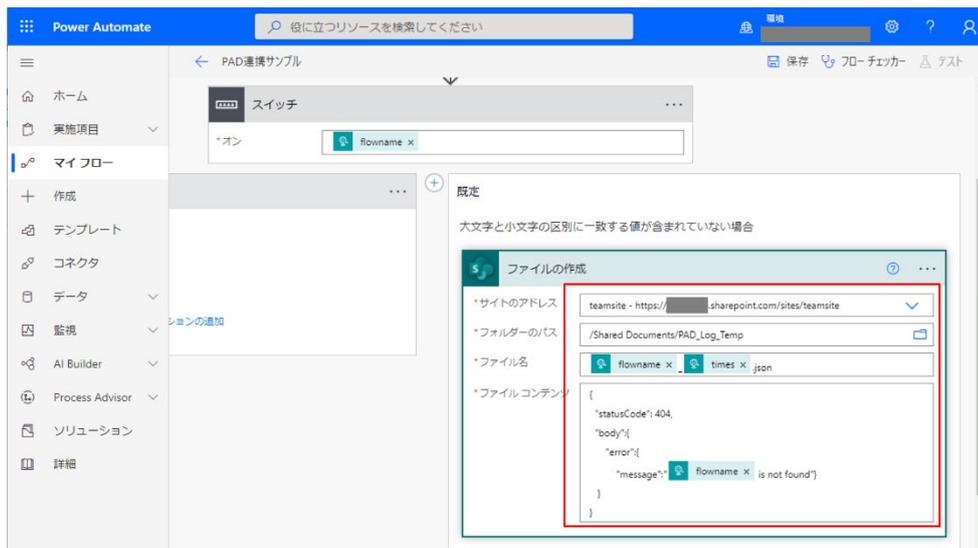
次に「ファイルコンテンツ」に以下の内容を記載すると、画面のような表示になります。

この設定により、要求された「flowname」パラメーターが指定外の場合のエラー応答が可能になります。

```

{
  "statusCode": 404,
  "body": {
    "error": {
      "message": "@{triggerBody()?['flowname']} is not found"
    }
  }
}

```



- (1) 「スイッチ」で分岐された「ケース」にある「アクションの追加」を選択します。

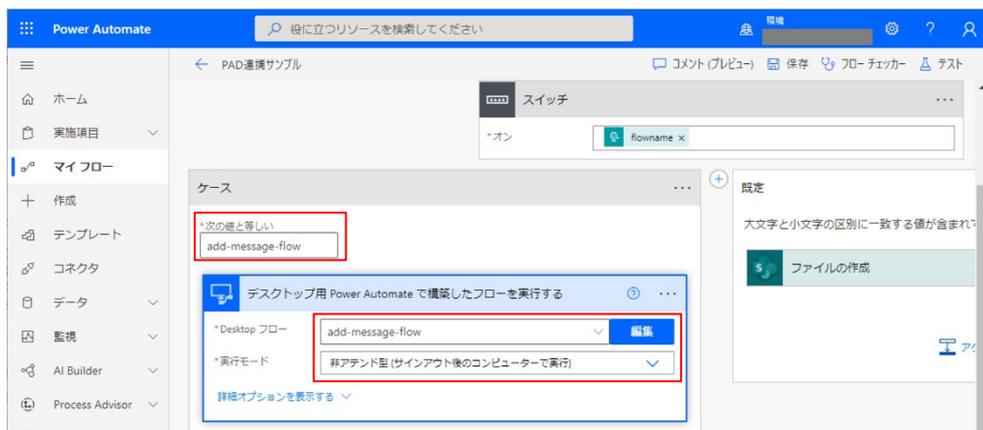


- (m) 新しいアクションとして「操作を選択してください」がフローに追加されます。検索窓に「power automate desktop」と入力します。次に「アクション」の「Power Automate で構築したフローを実行する」を選択します。



- (n) 「Power Automate で構築したフローを実行する」の設定が開きます。「Desktop フロー」に、実行する Desktop フローを選択します。「実行モード」に「非アテンド型・サインインしなくてもバックグラウンドで実行する」を選択します。次に「次の値と等しい」の入力個所に「Desktop フロー」で表示されている名称※2を入力します(今回のケースでは、add-message-flow となります)。

※2：本設定値は、JP1/AJS3 から Power Automate for desktop 連携ツールを呼ぶ際のパラメーターとなります。



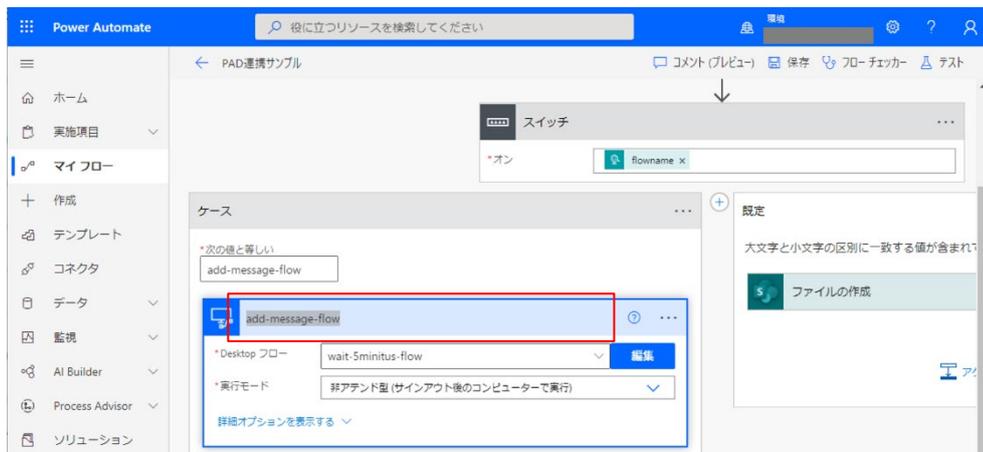
- (o) 「Power Automate で構築したフローを実行する」の「…」を選択し、「名前の変更」を選択します。



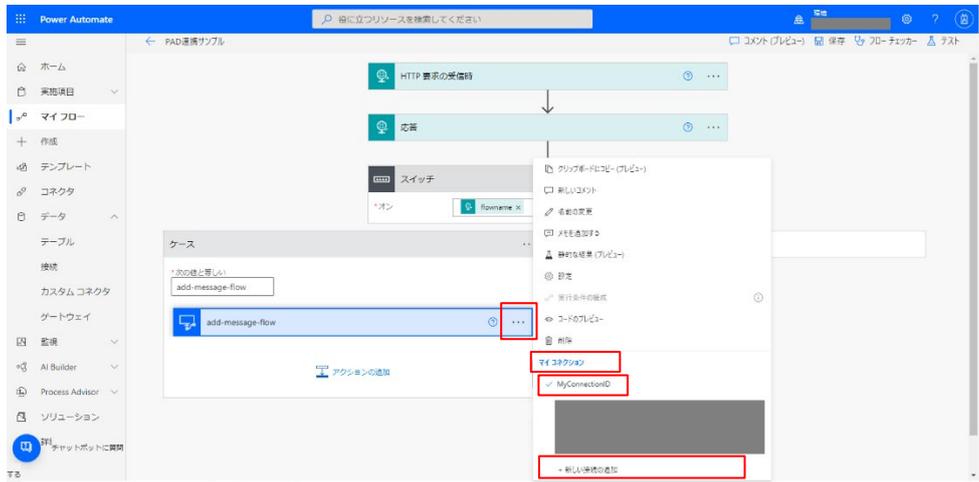
- (p) 「Desktop フロー」と同じ名称※3を入力します。

※3：この名称は、次に設定する応答アクションで使用します。

(半角スペースは、「_」に変更してください。)

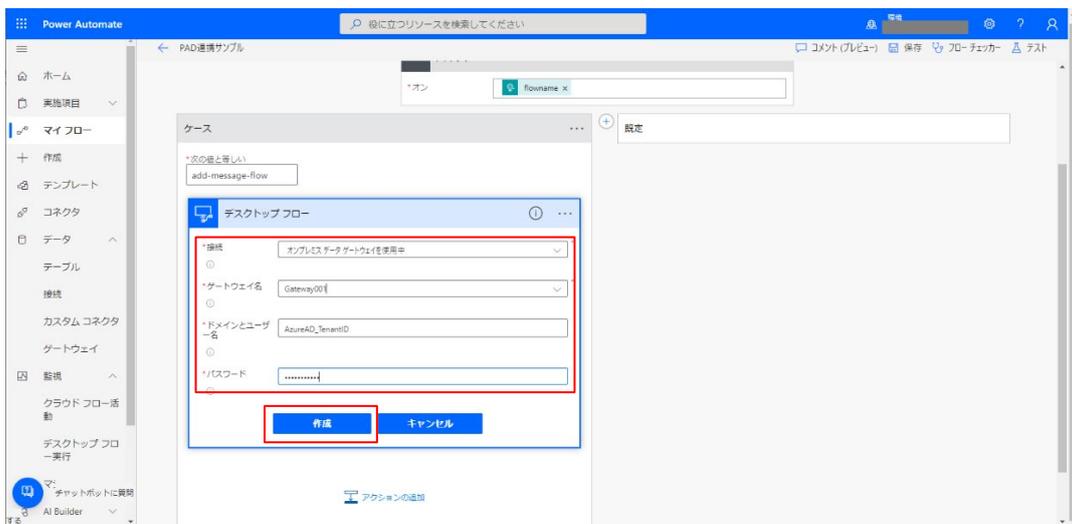


- (q) 接続先の設定を行います。「…」を選択します。既に「マイコネクション」に該当の接続情報が表示され、選択されている場合は手順(v)までお進みください。また、接続情報が存在しない場合は、「新しい接続の追加」を選択します。

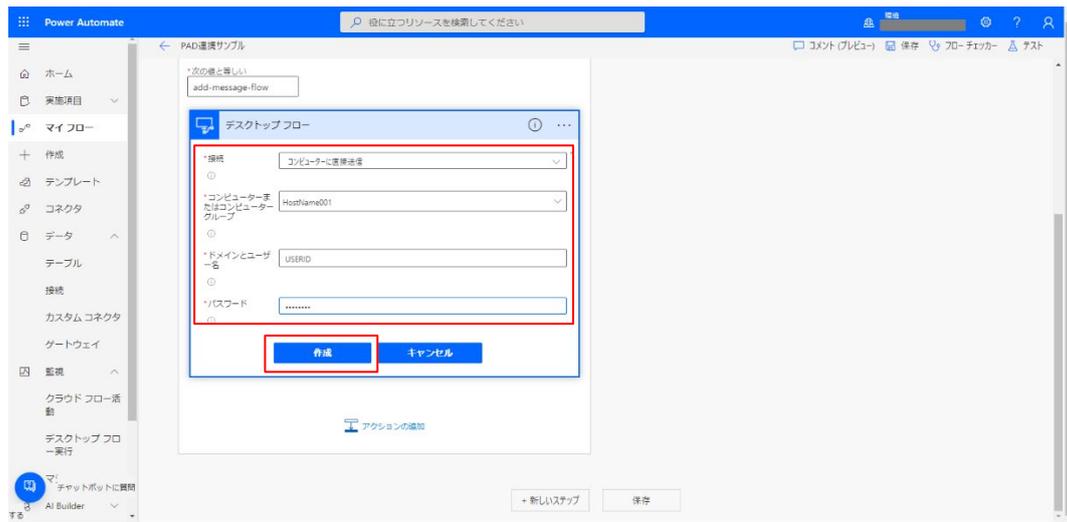


- (r) 「接続の追加」の設定が開きます。オンプレミス データ ゲートウェイを利用する場合は、手順(s)を行ってください。オンプレミス データ ゲートウェイを利用しない場合は、手順(t)までお進みください。

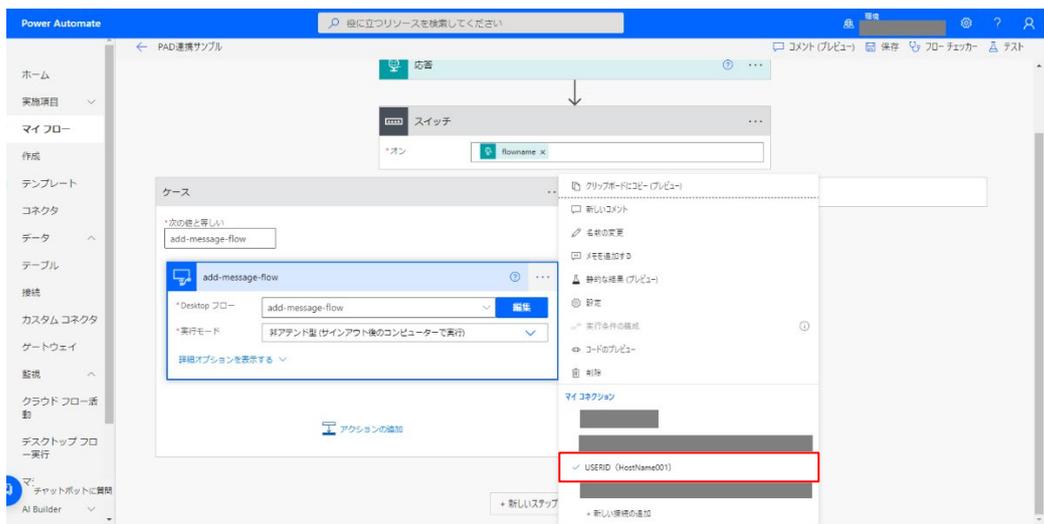
- (s) 「接続」に「オンプレミスデータゲートウェイを使用中」、「ゲートウェイ名」に「該当のゲートウェイ」を選択、「ドメインとユーザ名」に「Azure AD テナント内のアカウント」、「パスワード」に「Azure AD テナント内のアカウントのパスワード」を入力し、「作成」ボタンを押下します。



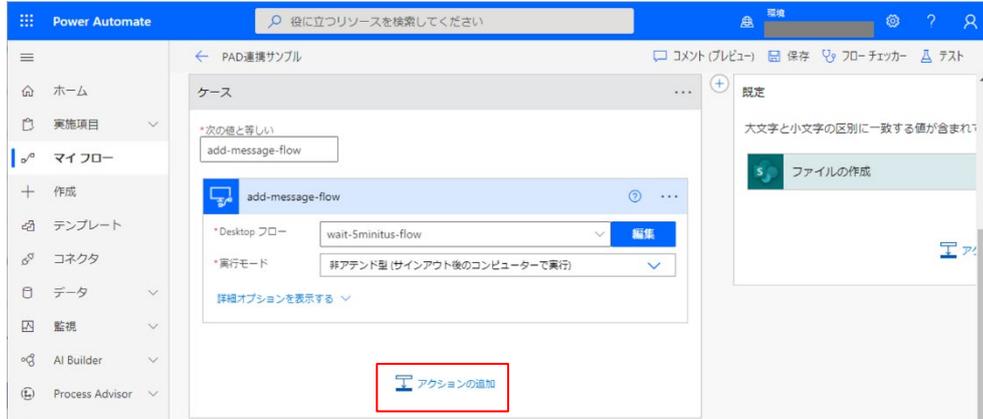
- (t) 「接続」に「コンピュータに直接送信」、「コンピューターまたはコンピューターグループ」に「該当のロボット実行端末名」を選択、「ドメインとユーザ名」に「該当のロボット実行端末にログインするユーザ名」、「パスワード」に「該当のロボット実行端末にログインするパスワード」を入力し、「作成」ボタンを押下します。



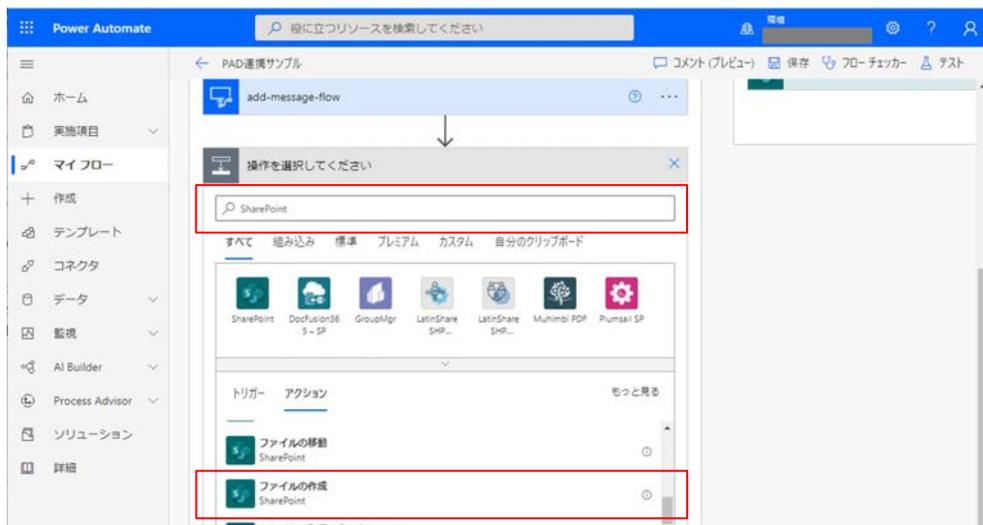
- (u) 「作成」ボタン押下後、再度「…」を選択し、一つ前の手順で入力した内容にチェックが付いていることを確認します。



- (v) 「スイッチ」で分岐された「ケース」にある「アクションの追加」を選択します。



- (w) 新しいアクションとして「操作を選択してください」がフローに追加されます。検索窓に「SharePoint」と入力し、「ファイルの作成」を選択します。



- (x) 「ファイルの作成 2」の設定が開きます。「サイトのアドレス」に「カスタム値の入力」を選択し、アクセス可能な SharePoint Online チームサイトの URL を入力します (この URL については(k)で指定した URL と同じ URL を指定してください)。

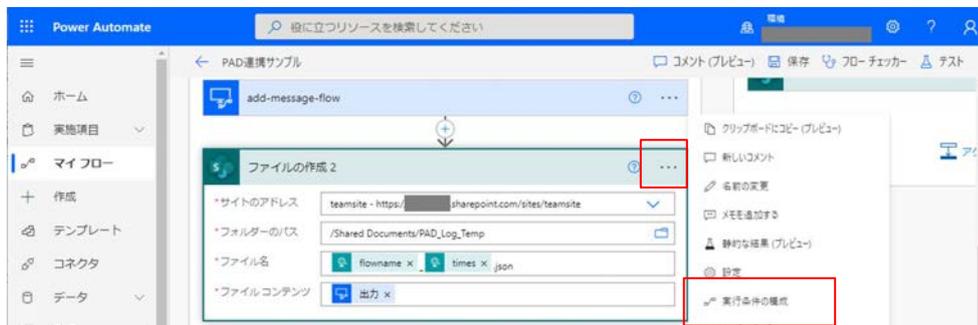
次に「フォルダーのパス」の右端にある「ピッカーの表示」を選択し、対象の SharePoint Online チームサイトにある、データ連携用フォルダを選択してください (このフォルダについては(k)で指定したフォルダと同じフォルダを指定してください)。

次に「ファイル名」の箇所を選択し、

「@{triggerBody()?['flowname']}_{triggerBody()?['times']}.json」を入力すると、画面のようなファイル名の表示になります。次に「ファイルコンテンツ」の箇所を選択し、「@{outputs('[(p)で指定した名称(※3)]}」を入力します。「ファイルコンテンツ」の入力箇所が、青い Desktop フローのアイコンに変わっていることを確認します。



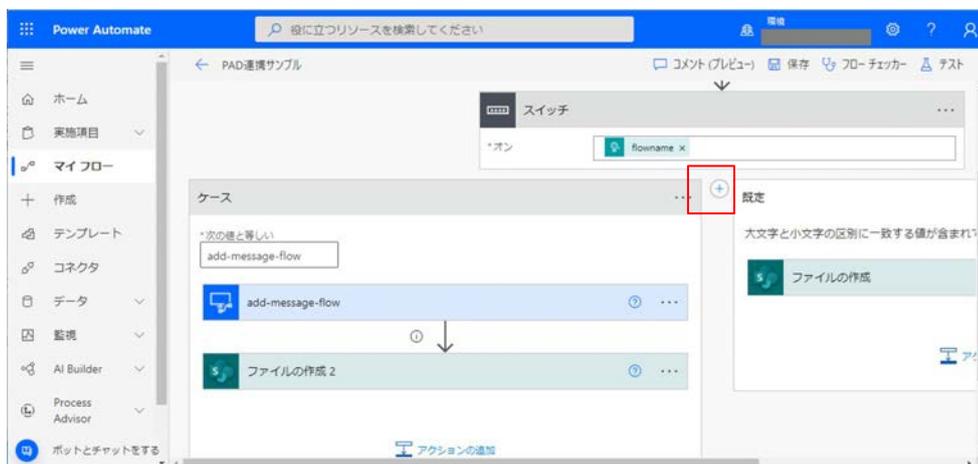
(y) 「ファイルの作成 2」の「…」を選択し、「実行条件の構成」を選択します。



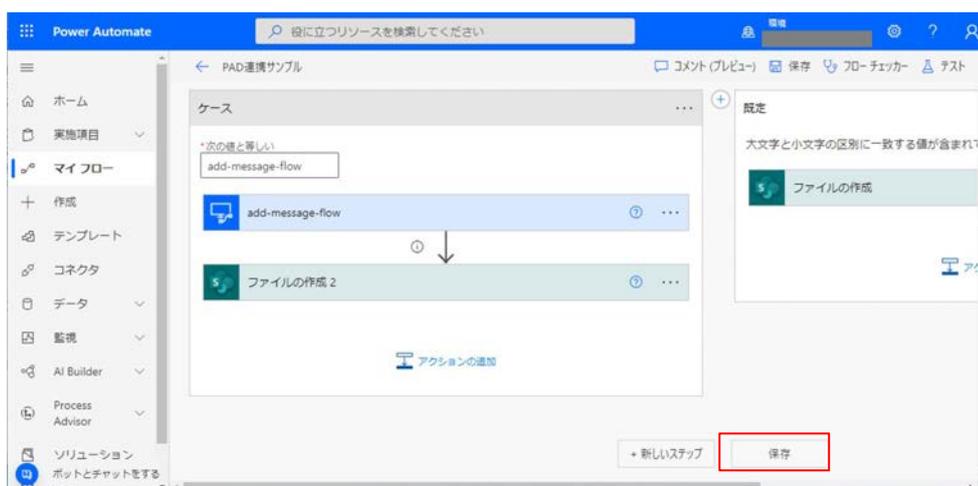
(z) 「ファイルの作成 2 が実行されるタイミング」の設定が開きます。「に成功しました」「に失敗しました」「がスキップされます」「がタイムアウトしました」すべてにチェックを入れ、「完了」を選択します。



(aa) 実行する Desktop フローが複数ある場合、「ケース」の右側にある「+」を選択し、ケースを追加します。その後、(1)に戻りそれ以降、同様の設定を行います。(クラウドフローの保存後、クラウドフローの編集でも実施可能です)



以上の設定が完了したら「保存」を選択して、クラウドフローを保存します。



5. 2. 3. 2. Power Automate クラウドフローHTTP 要求受信時トリガーURL の取得

Power Automate のクラウドフローを作成・保存後、クラウドフローの HTTP 要求受信時トリガーの URL を取得します。

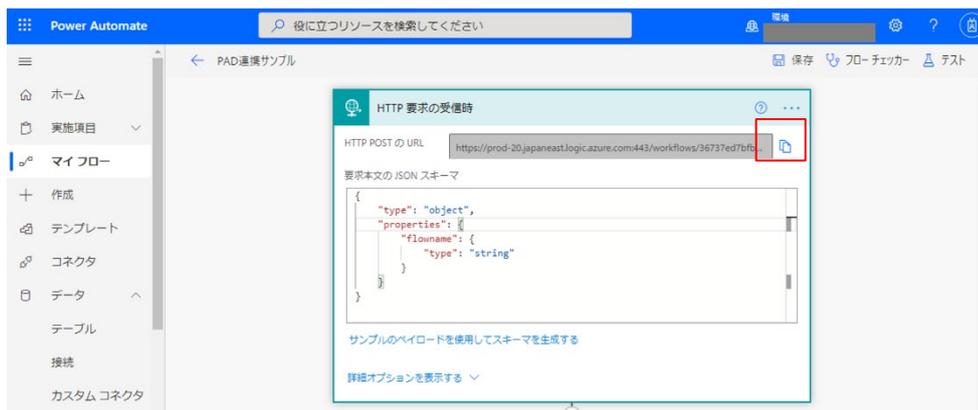
(1) HTTP 要求の受信時トリガーの要求 URL の取得をします。

- (a) <https://japan.flow.microsoft.com/ja-jp/> へブラウザからアクセスし、Power Automate を開発するユーザーでサインインし、「マイフロー」を選択し「5. 2. 3. 1 Power Automate for desktop を実行するための、Power Automate クラウドフロー作成手順上の」で作成したフローの「編集」を選択します。



- (b) 「HTTP 要求の受信時」を選択します。「HTTP POST の URL」が自動生成されていますので、赤枠のコピーボタンを選択し、URL※4 をテキストメモに保存します。

※4：この URL が Power Automate for desktop 連携ツールでの、\$BaseUrl パラメーター設定値になります。



5.2.4. Power Automate for desktop 連携ツールの構成

Power Automate for desktop 連携ツールの提供物（ディレクトリ・ファイル）の構成を以下に示します。

表 5.4 Power Automate for desktop 連携ツールのディレクトリ・ファイル構成

#	ディレクトリ／ファイル	説明
1	PA_FlowExec	Power Automate for desktop 連携ツールディレクトリ(名称、ディレクトリ位置は任意)
2	PAFlowExecutor.bat	Power Automate for desktop 連携コマンド
3	PAFlowExecutor.ps1	Power Automate for desktop 連携内部プログラムファイル
4	PAConfig.ps1	Power Automate for desktop 連携構成定義ファイル
5	log	Power Automate for desktop 連携ツール実行ログ出力ディレクトリ(名称、ディレクトリ位置は任意)
6	download-files	Power Automate for desktop 実行結果取得データ格納ディレクトリ(名称、ディレクトリ位置は任意)

5.2.5. Power Automate for desktop 連携ツール インストール手順

5.2.5.1. Power Automate for desktop 連携ツールの配置

Power Automate for desktop 連携ツールを配置する手順を以下に示します。

- (a) JP1/AJS3 管理サーバーへログインします。
- (b) 「表 5.4 Power Automate for desktop 連携ツールのディレクトリ・ファイル構成」の「PA_FlowExec」「log」「download-files」ディレクトリを任意のディレクトリに作成します。
- (c) 「表 5.4 Power Automate for desktop 連携ツールのディレクトリ・ファイル構成」の「PA_FlowExec」に該当するディレクトリへ、次に示す「PAFlowExecutor.bat」「PAFlowExecutor.ps1」「PAConfig.ps1」スクリプトをファイルとして作成します。

<PAFlowExecutor.bat>

```
@echo off

REM スクリプト実行
echo Start Script.
PowerShell -ExecutionPolicy RemoteSigned -File "%~dp0\PAFlowExecutor.ps1" %*

REM 結果を出力
echo End Script. Return Code: %ERRORLEVEL%

REM 常に正常で終了
exit %ERRORLEVEL%

@echo on
```

<PAFlowExecutor.ps1>

```
Param (
    [switch] $start,
    [string] $a_flowname = ""
)

#-----
# 共通ファイル読込
# 共通モジュール名
$ConfigModuleFileName = "PAConfig.ps1"

# カレントディレクトリパス
$CurrentDirPath = Split-Path $MyInvocation.MyCommand.Path -Parent

# 定数モジュール
. (Join-Path ${CurrentDirPath} ${ConfigModuleFileName})

#-----
# 定数
$ExitCodeSuccess = 0
$ExitCodeFlowError = 1
$ExitCodeFlowTimeout = 2
$ExitCodeSystemError = 255
```

```

# ログ出力ファイル
# ログファイルフォーマット
$FileNameDate = Get-Date -Format "yyyyMMdd"
$PowerAutomateScriptLogFormat = "{0}_PAFlowExec.log"
$PAFlowExeLogFile = if ($OutputLogFilePath) { Join-Path $OutputLogFilePath ($PowerAutomateScriptLogFormat) -f ${FileNameDate} } else { Join-Path $CurrentDirPath ($PowerAutomateScriptLogFormat) -f ${FileNameDate} }

# ログレベル
$LogLevelDebug = "Debug"
$LogLevelInfo = "Info"
$LogLevelWarn = "Warn"
$LogLevelError = "Error"

# ログメッセージフォーマット
$LogMessageFormat = "{0} {1}: {2}"

# ログメッセージ日時フォーマット
$LogDateTimeFormat = "yyyy/MM/dd HH:mm:ss"

# ログレベルの桁数
$LogLevelLength = 5
$LogLevelPadChar = " "

# Header情報 - ContentType
set-variable -name ContentType -value "application/json" -option constant
# Header情報 - Accept
set-variable -name Accept -value "application/json" -option constant

#-----
# グローバル変数
# 実行プロセス名
$FlowName = ""

#-----
# 関数
Function Write-Log($LogLevel, $Message, $StdErr) {
    if (($LogLevel -ne $LogLevelDebug) -or ($debug -and ($LogLevel -eq $LogLevelDebug))) {
        $logMessage = ""

        if ($PAFlowExeLogFile) {
            # 出力メッセージ
            if ($Message) {
                $logMessage = $Message
            }
            else {
                $logMessage = $input | Out-String
            }

            # ログ出力
            Out-File -FilePath $PAFlowExeLogFile -InputObject (Get-LogMessage
$LogLevel $logMessage) -Encoding Default -Append

            # 標準エラーへの出力
            if ($StdErr) {
                $Host.UI.WriteLine((Get-LogMessage $LogLevel $logMessage))
            }
        }
    }
}

Function Get-LogMessage($LogLevel, $Message) {
    return ($LogMessageFormat -f (Get-Date -Format $LogDateTimeFormat),

```

```

${LogLevel}.PadLeft(${LogLevelLength}, ${LogLevelPadChar}), ${Message})
}

function Test-Parameters() {
    # 実行モードチェック
    if (-not ${start}) {
        # 実行モードが指定されていない場合
        throw "Parameter '-start' is required."
    }

    # プロセス名チェック
    if (${start} -and -not ${a_flowname}) {
        # 空の場合
        throw "Invalid flowname. (flowname is blank.)"
    } else {
        $script:FlowName = ${a_flowname}
    }
}

function Set-ProxyCredential
{
    $securePassword = ConvertTo-SecureString $ProxyPassword -AsPlainText -Force
    $script:proxyCredential = New-Object System.Management.Automation.PSCredential -ArgumentList
$ProxyUser, $securePassword
}

Function DeleteFile($FolderURLPath, $FileName) {
    $TargetFolderURL = ""
    # 対象フォルダ取得
    $objFolder = $Context.Web.GetFolderByServerRelativeUrl($FolderURLPath)
    $Context.Load($objFolder)
    $Context.ExecuteQuery()

    # フォルダ配下のファイルコレクションを取得
    $Files = $objFolder.Files
    $Context.Load($Files)
    $Context.ExecuteQuery()
    $flag = "false"
    # ファイルが存在した場合、探してるファイルかどうかをチェックし、対象であれば削除
    If ($Files.count -ge 1) {
        foreach($File in $Files) {
            If ($File.Name -eq $FileName) {
                $File.DeleteObject()
                $Context.ExecuteQuery()
                $flag = "true"
                break
            }
        }
    }

    return $flag
}

Function FindFile($FolderURLPath, $FileName) {
    $TargetFolderURL = ""

    # 対象フォルダ取得
    $objFolder = $Context.Web.GetFolderByServerRelativeUrl($FolderURLPath)
    $Context.Load($objFolder)
    $Context.ExecuteQuery()
}

```

```

# フォルダ配下のファイルコレクションを取得
$Files = $objFolder.Files
$Context.Load($Files)
$Context.ExecuteQuery()
$flag = "false"
# ファイルが存在した場合、探してるファイルかどうかをチェック
If ($Files.count -ge 1){
    foreach($File in $Files){
        If ($File.Name -eq $FileName){
            $flag = "true"
            break
        }
    }
}

return $flag
}

#-----
# メイン処理

# Sharepoint Onlineに繋ぐためのランタイムロード
[System.Reflection.Assembly]::LoadWithPartialName("Microsoft.SharePoint.Client") > $null
[System.Reflection.Assembly]::LoadWithPartialName("Microsoft.SharePoint.Client.Runtime") > $null

$ExitCode = 0

try {
    Write-Log ${LogLevelInfo} "Start."

    # パラメータチェック
    Test-Parameters

    $filedatemillis = (Get-Date).ToString("yyyy-MM-dd-HHmss.ffff")

    # Proxy利用判定
    if ($ProxyEnable -eq "true") {
        # ProxyCredentialの設定
        Set-ProxyCredential
    }

    $strigFlowReqBody = @{
        'flowname' = $flowname;'times' = $filedatemillis
    } | ConvertTo-Json

    Write-Log ${LogLevelInfo} ("Start FlowName: FlowName: {0}" -f ${flowname})
    try {
        # SPO認証処理
        $pwd = convertto-securestring $SpoPassword -AsPlainText -Force
        $credential = New-Object Microsoft.SharePoint.Client.SharePointOnlineCredent-
entials($SpoUserName, $pwd)
        $Context = New-Object Microsoft.SharePoint.Client.ClientContext($SpoSiteURL)
        $Context.Credentials = $credential

        $objWeb = $Context.Web
        $Context.Load($objWeb)
        $Context.ExecuteQuery()

        # SPOの過去に出力されている同名のファイルが存在する場合削除
        $TargetFolderURL = $SpoSiteURL + $SpoStaticURL
        $TargetFileName = $flowname + "_" + $filedatemillis + ".json"
        $Ret = DeleteFile $TargetFolderURL $TargetFileName

        Write-Log ${LogLevelInfo} ("pre SPO File Delete: FolderURL: {0}" -f ${TargetFolderURL})
    }
}

```

```

Write-Log ${LogLevelInfo} ("pre SPO File Delete: FileName: {0}" -f ${TargetFileName})
Write-Log ${LogLevelInfo} ("pre SPO File Delete: RetCode: {0}" -f ${Ret})

if (![string]::IsNullOrEmpty($proxyCredential)) {
    # Proxy認証あり
    $flowResponse = Invoke-RestMethod -Proxy $ProxyUrl -ProxyCredential
$proxyCredential -Method "POST" -Uri $BaseUrl -Body $trigFlowReqBody -ContentType $ContentType
}
else {
    # Proxy認証なし
    $flowResponse = Invoke-RestMethod -Method "POST" -Uri $BaseUrl -Body
$trigFlowReqBody -ContentType $ContentType
}

# 実行受付StatusCodeの判定
if ($flowResponse.statusCode -eq 200) {
    # 実行受付正常
    Write-Log ${LogLevelInfo} ("request StatusCode: {0}" -f $flowResponse.status-
Code)

    Write-Log ${LogLevelInfo} ("request Response-headers: {0}" -f $flow-
Response.headers)

    # SPOに実行結果ファイルが作成されるまでポーリング
    $findflg = "false"
    Write-Log ${LogLevelInfo} ("SPO file search: FolderURL: {0}" -f ${Target-
FolderURL})

    Write-Log ${LogLevelInfo} ("SPO file search: FileName: {0}" -f ${TargetFile-
Name})

    for ($i=0; $i -lt $SpoCheckCount; $i++){
        $Ret = FindFile $TargetFolderURL $TargetFileName
        Write-Log ${LogLevelInfo} ("SPO file search: RetCode: {0}" -f
${Ret})

        if ($Ret -eq "true") {
            $findflg = "true"
            break
        }
        Sleep $SpoCheckInterval
    }

    if ($findflg -eq "true") {
        Sleep 3

        # ダウンロードフォルダがなければ、作成
        If(-Not(Test-Path $OutputFolderPath)) {
            New-Item $OutputFolderPath -ItemType Directory | Out-
Null

            Write-Log ${LogLevelInfo} "SPO download folder created."
        }

        # SPOから実行結果ファイルをダウンロード
        $FileFullPath = $TargetFolderURL + "/" + $TargetFileName

        $intFileCurrentPathStart = $FileFullPath.IndexOf($SpoStaticURL) +
$SpoStaticURL.Length

        $intFileCurrentPathEnd = $FileFullPath.Length

        $FilePath = $FileFullPath.Substring($intFileCurrentPath-
Start, ($intFileCurrentPathEnd - $intFileCurrentPathStart))

        $intFileNameStart = $FileFullPath.LastIndexOf('/') + 1
        $intFileNameEnd = $FileFullPath.Length

        $FileName = $FileFullPath.Substring($intFile-
NameStart, ($intFileNameEnd - $intFileNameStart))

```

```

$SpoFile = $ObjWeb.ServerRelativeUrl + $SpoStaticURL + $FilePath
$SaveFile = $OutputFolderPath + $FileName

を開き、データコピーを行う
rect($Context, $SpoFile)
Mode)::Create)

Raw)
内容を変更する。

$SpoJson.statusCode) $true
$SpoJson.headers) $true
$SpoJson.statusCode) $true
$SpoJson.headers) $true
f $SpoJson.body.error.code) $true
{0}" -f $SpoJson.body.error.message) $true

} else {
# 監視時間タイムアウト
Write-Log ${LogLevelError} ("StatusCode: {0}" -f "504") $true
Write-Log ${LogLevelError} ("Response-Error-Message: {0}" -f "SPO
file search timed out.") $true
$ExitCode = ${ExitCodeFlowTimeout}

# SPOに出力されたJSONファイルの削除
$Ret = DeleteFile $TargetFolderURL $TargetFileName
Write-Log ${LogLevelInfo} ("SPO File Delete: FolderURL: {0}" -f ${Target-
FolderURL})
Write-Log ${LogLevelInfo} ("SPO File Delete: FileName: {0}" -f ${TargetFile-
Name})
Write-Log ${LogLevelInfo} ("SPO File Delete: RetCode: {0}" -f ${Ret})

} else {
Write-Log ${LogLevelError} ("StatusCode: {0}" -f $flowResponse.statusCode)
$true
Write-Log ${LogLevelError} ("Response-headers: {0}" -f $flowResponse.headers)
$true
Write-Log ${LogLevelError} ("Response-Error-Code: {0}" -f $flow-
Response.body.error.code) $true
Write-Log ${LogLevelError} ("Response-Error-Message: {0}" -f $flow-
Response.body.error.message) $true
$ExitCode = ${ExitCodeFlowError}
}
} catch {
if ($_.Exception.Response -eq $null) {

```

```

        Write-Log ${LogLevelError} ("Response: {0}" -f $_.Exception) $true
    } else {
        $r = $_.Exception.Response
        $rs = $r.GetResponseStream()
        $rs.Position = 0
        $sr = [System.IO.StreamReader]::new($rs)
        $res = $sr.ReadToEnd()
        $sr.Close()
        Write-Log ${LogLevelError} ("StatusCode: {0}" -f $r.StatusCode.Value__) $true
        Write-Log ${LogLevelError} ("Response: {0}" -f $res) $true
    }
    $ExitCode = ${ExitCodeSystemError}
}
} catch {
    # 例外発生
    ${error}[0] | Write-Log ${LogLevelError} ${error}[0] $true
    $ExitCode = ${ExitCodeSystemError}
} finally {
}

Write-Log ${LogLevelInfo} ("Finish. ReturnCode {0}" -f $ExitCode)
exit $ExitCode

```

<PAConfig.ps1>

```

#####
# PowerAutomate連携スクリプト（構成定義）
#####
# PowerAutomateBaseURL
$BaseUrl = "https://"

# ProxySettings
$ProxyEnable = "false"
$ProxyUrl = ""
$ProxyUser = ""
$ProxyPassword = ""

# SPO Settings
$outputFolderPath = "C:\Program Files (x86)\Hitachi\JP1RPA\PA_FlowExec\download-files"

$SPOUserName = ""
$SPOPassword = ""
$SPOSiteURL = "https://"
$SPOStaticURL = ""

# SPO File Check Interval (s)
$SPOCheckInterval = 60

# SPO File Check Count
$SPOCheckCount = 120

# LogInformation
$outputLogFilePath = "C:\Program Files (x86)\Hitachi\JP1RPA\PA_FlowExec\Log"

```

5.2.6. Power Automate for desktop 連携ツール セットアップ手順

5.2.6.1. Power Automate for desktop 連携ツール 構成定義

Power Automate for desktop 連携ツールを動作させるための各パラメーター設定手順を以下に示します。

- (a) JP1/AJS3 管理サーバーへログインします。
- (b) Power Automate for desktop 連携ツールディレクトリ配下にある「PAConfig.ps1」をエディタで開き、ユーザー可変値を設定します。

表 5.5 Power Automate for desktop 連携ツール 構成定義 ユーザー可変値

#	ユーザー可変値	必須／任意	内容
1	\$BaseUrl	必須	Power Automate クラウドフローの HTTP 要求受信時トリガーURL を指定します。 (「5.2.3.1(1)(b)」で取得した URL(※4)を指定します。)
2	\$ProxyEnable	必須	Proxy 設定有無を” true” または” false” で指定します。
3	\$ProxyUrl	任意	Proxy 設定有無に” true” を指定した場合に、Proxy の URL を指定します。
4	\$ProxyUser	任意	Proxy 設定有無に” true” を指定した場合に、Proxy の認証ユーザー名を指定します。
5	\$ProxyPassword	任意	Proxy 設定有無に” true” を指定した場合に、Proxy の認証ユーザーパスワードを指定します。
6	\$OutputFolderPath	必須	SharePoint Online から PowerAutomate for desktop の実行結果をダウンロードするフォルダを指定します。
7	\$SpoUserName	必須	SharePointOnline のチームサイトへアクセス可能なユーザー名を指定します。 指定するユーザーは、多要素認証が無効になっている必要があります。
8	\$SpoPassword	必須	SharePointOnline のチームサイトへアクセス可能なユーザー名のパスワードを指定します。
9	\$SpoSiteURL	必須	SharePointOnline のチームサイトの URL を指定します。(「5.2.3.1(1)(k)」で指定したサイトの URL と同一になります)
10	\$SpoStaticURL	必須	SharePointOnline のチームサイトのフォルダパスを指定します。(「5.2.3.1(1)(k)」で指定したフォルダのパスと同一になります)

5.2.7. Power Automate for desktop 連携ツール アンインストール手順

5.2.7.1. Power Automate for desktop 連携ツールの削除

- (a) JP1/AJS3 管理サーバーへログインします。
- (b) 「5.2.5.1 Power Automate for desktop 連携ツールの配置」でコピーした「PA_FlowExec」ディレクトリを削除します。

5.3. 稼働確認

5.3.1. Power Automate for desktop 連携ツールを利用した JP1/AJS3 と Power Automate for desktop との連携方法

5.3.1.1. 実行

JP1/AJS3 – View の[ジョブネットエディタ]ウィンドウで、Power Automate for desktop 連携ツールを実行するジョブを作成します。

ジョブの定義で指定する Power Automate for desktop 連携ツールの「実行形式」、「実行ファイル」、「オプション」について以下に示します。

(実行形式)

```
Power Automate for desktop 連携ツールのインストールフォルダ¥PAFlowExecutor.bat  
-start 「実行する Desktop フロー名」
```

「実行する Desktop フロー名」は、「5.2.3.1 (1) (n)」にて設定した値(※2)を使用します。

(実行ファイル)

Power Automate for desktop 連携ツールのインストールフォルダに存在する「PAFlowExecutor.bat」ファイルをフルパスで指定します。

(オプション・引数)

表 5.6 オプション・引数説明

#	引数	説明	必須	備考
1	-start	プロセスを実行するモードで連携ツールを起動します。	○	未指定の場合、ジョブは異常終了します。
2	「実行する Desktop フロー名」	実行する Desktop フロー名を指定します。	○	未指定の場合、ジョブは異常終了します。

【凡例】 ○ : 必須

※ オプションの指定例

```
PAFlowExecutor.bat -start add-message-flow
```

(JP1/AJS3 定義例)

詳細定義-[PC Job]

ユニット名 add-message-flow

コメント

実行エージェント

定義 転送ファイル 属性

実行ファイル名 hitachi\JP1RPA\PA_FlowExec\PAFlowExecutor.bat

パラメーター -start add-message-flow

環境変数

環境変数ファイル名

ワークパス

実行優先順位 なし

標準入力ファイル名

標準出力ファイル名 追加書き

標準エラー出力ファイル名 追加書き

終了判定 判定結果 しきい値による判定
警告しきい値 異常しきい値 0

異常終了時リトライ しない する

終了コード 以上 以下

最大リトライ回数 1 回

リトライ間隔 1 分

実行時のユーザー

OK キャンセル ヘルプ

5.3.1.2. 結果確認

- JP1/AJS3 - View ウィンドウ（メイン画面）で、実行したジョブネットを右クリックし、「ジョブネットモニタ」→「結果」の順で選択します。
 - 「ジョブネットモニタ」が表示されますので、ジョブを右クリックして「詳細情報」を選択します。
 - 「モニタ詳細」が表示されますので、Power Automate for desktop 連携ツールの結果コードとなる「終了コード」を確認します。
- また、詳細な情報は、「モニタ詳細」の「実行結果詳細」の詳細ボタンを押下して確認します。

(モニタ詳細)

モニタ詳細 - [PCジョブ]

ユニット名	/PowerAutomate/PA-JOBNET001/add-message-flow
状態	正常終了
計画一時変更	なし
保留	定義に従う (保留しない)
開始日時	2021/07/26 14:22:36
終了日時	2021/07/26 14:24:02
遅延状態	なし
終了遅延日時	なし
終了遅延	なし
ジョブ登録日時	2021/07/26 14:22:35
ジョブ番号	301
終了コード	0
実行ホスト名	
待ち合わせ	-
リトライ状態	なし
リトライ実行回数	0
リトライ開始日時	なし
リトライ登録日時	なし
実行結果詳細	詳細...

定義編集... 閉じる ヘルプ

Power Automate Desktop 連携ツールの実行結果コード(※5)

実行結果詳細画面の表示

(実行結果詳細)

実行結果詳細

```
2021/07/26 14:24:02 Info: StatusCode: 200
2021/07/26 14:24:02 Info: Response-headers: @[Pragma=no-cache; Vary=Accept-Encoding; x-ms-run-id=cea5b200-d1ed-eb11-94ef-0022488213ab; x-ms-tr
```

実行結果詳細

閉じる

Power Automate または、Power Automate Desktopからの応答メッセージ

※5 : Power Automate for desktop 連携ツールの実行結果コードを以下に示します。

表 5.7 実行結果コード説明

#	実行結果コード	説明
1	0	正常終了
2	1	Power Automate for desktop で異常終了
3	2	Power Automate for desktop 実行指示後の応答タイムアウト
4	255	その他異常終了

5.4. トラブルシューティング

5.4.1. 異常時の対処

Power Automate for desktop 連携ツールの実行結果コードが 0 以外の場合は、JP1/AJS3 - View の「実行結果詳細」または、Power Automate for desktop 連携ツールが出力しているログからエラーメッセージを確認して、問題を対処してください。

<実行結果コード : 1 の例>

```
2021/07/26 14:31:00 Info: Start.
2021/07/26 14:31:00 Info: Start FlowName: FlowName: error-flow
2021/07/26 14:32:17 Error: StatusCode: 400
2021/07/26 14:32:17 Error: Response-headers: @[Pragma=no-cache; x-ms-run-id=ca3ea2a9-d2ed-eb11-94ef-0022486213ab; x-ms-request-id=japaneast:0770d410-c6d1-4784-b5ec-d78a94a5c7de; x-ms-correlation-request-id=0770d410-c6d1-4784-b5ec-d78a94a5c7de; x-ms-flow-mobile-ios-version=1.3.0; x-ms-flow-routing-request-id=JAPANEAST:20210726T053217Z:0770d410-c6d1-4784-b5ec-d78a94a5c7de; Strict-Transport-Security=max-age=31536000; includeSubDomains; X-Content-Type-Options=nosniff; x-ms-connection-gateway-object-id=67a421c1-8d97-43c6-a264-ee5620a66550; Timing-Allow-Origin=*; x-ms-apidhub-cached-response=true; Cache-Control=no-cache; Date=Mon, 26 Jul 2021 05:32:16 GMT; Content-Length=192; Content-Type=application/json; Expires=-1]
2021/07/26 14:32:17 Error: Response-Error-Code: File.InvalidDestinationFolderError
2021/07/26 14:32:17 Error: Response-Error-Message: Problem while executing action 'Copy'. Destination folder C:¥temp¥work¥nonfile.txt doesn't exist
2021/07/26 14:32:17 Info: Finish. ReturnCode 1
```

エラーメッセージ
PAD の実行エラー内容が表示されている

<実行結果コード : 255 の例>

```
2021/07/26 14:33:29 Info: Start.
2021/07/26 14:33:29 Info: Start FlowName: FlowName: wait-5minites-flow
2021/07/26 14:35:38 Error: StatusCode: 504
2021/07/26 14:35:38 Error: Response: [{"error":{"code":"ResponseTimeout","message":"The server did not receive a timely response from the upstream server. Request tracking id '08585743292758642682471841777CU24'."}}]
2021/07/26 14:35:38 Info: Finish. ReturnCode 255
```

エラーメッセージ
今回はタイムアウトの例