

自動化を開発する

PL500 試験では「自動化を開発する (45-50%)」の範囲でクラウドフローとデスクトップフローの構築、外部サービスとの連携、ロジックや例外処理の実装、カスタムコネクタの開発、インフラ管理、テストと構成管理が出題されます。

重要なのは **Power Automate** を用いた実践的な開発スキルと、運用に耐える堅牢な自動化設計の理解です。

1. クラウドフローを開発する

1.1. デスクトップフローを呼び出すクラウドフローを開発する

1.1.1. 概要

クラウドフローから「デスクトップフローの実行」アクションを使用することで、クラウドベースの処理からローカル環境での操作を自動化できます。

これにより、クラウドとオンプレミスの連携が可能になります。

1.1.2. 実行環境の要件

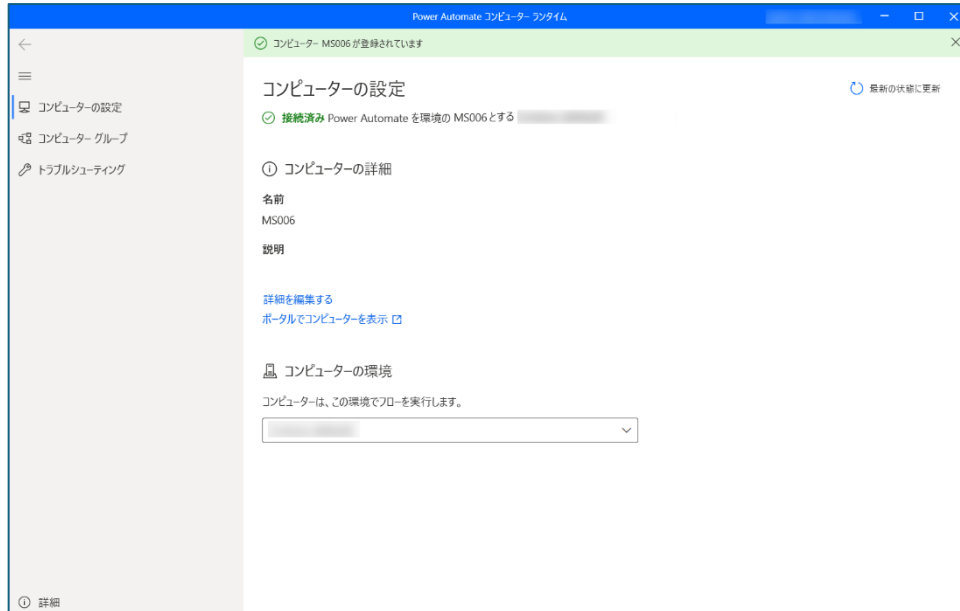
実行には主につぎのような前提条件を満たす必要があります。

- ①実行対象 PC には Windows 10/11 Pro, Enterprise, Server などの対応 OS が必要
(Home 版ではクラウドからの実行不可)
- ②実行対象 PC に以下がインストールおよび構成されていること
 - ・ Power Automate for desktop: デスクトップフローの作成・実行用
 - ・ Power Automate マシンランタイム: クラウドからマシンを接続・制御するため
- ③必要なライセンスがあること
 - ・ アテンド型 (Attended) 実行の場合: Power Automate Premium ユーザーライセンス
 - ・ 非アテンド型 (Unattended) 実行の場合: Power Automate Process ライセンス
- ④ Dataverse 環境が有効であること
クラウドフローとデスクトップフローの連携には、Dataverse ベースの環境が必要です

1.1.3. 実行対象のマシンの指定

デスクトップフローを実行するには、対象マシンが Power Automate に登録されている必要があります。複数のマシンを「コンピューターグループ」として管理することで、クラウドフローからグループ単位で実行対象を指定できます。

接続には「Power Automate コンピューターランタイム」▼を使用し、Azure Relay 経由でクラウドとマシンを接続します。

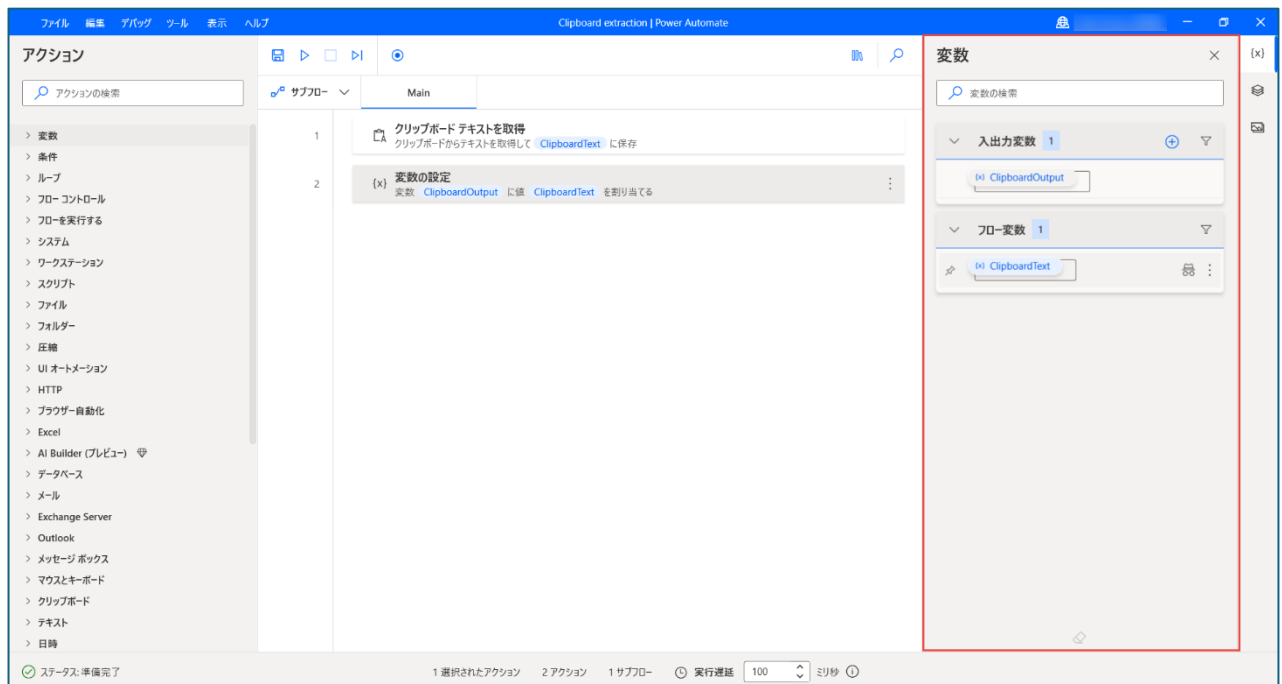


出典：[コンピューターの管理 - Power Automate | Microsoft Learn](#)

1.1.4. 入力パラメータの設定

クラウドフローからデスクトップフローに渡すデータ（例：ファイル名、検索条件など）をパラメータとして設定します。

▼入出力変数の定義



出典：[変数と変数ペインを管理する - Power Automate | Microsoft Learn](#)

1.1.5. 実行結果の取得

クラウドフローから呼び出したデスクトップフローの出力パラメータをクラウドフローに戻すことで、後続処理（通知、保存、条件分岐など）に活用できます。

具体的には以下のように操作します：

- ①デスクトップフローの「変数ペイン」で出力変数を作成します。
- ②クラウドフローの「デスクトップ用 Power Automate で構築されたフローを実行」アクションの「出力」として表示され、クラウドフロー内の処理に利用できます。

1.2. データの受け渡しや返却を含む子クラウドフローの開発と使用

1.2.1. 概要

子クラウドフローは、親フローから呼び出される再利用可能な処理単位です。
データの受け渡しや返却を通じて、フローの構造を整理し、保守性を高めることができます。

1.2.2. 子フローの作成方法

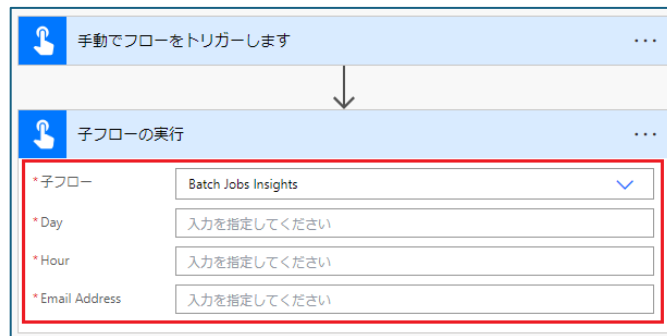
「手動トリガー」や「HTTP トリガー」を使って子フローを作成し、親フローから「フローの呼び出し」アクションで実行します。

両方のフローは同じソリューション内に作成する必要があります。

1.2.3. 入力パラメータの定義

子フロー側で受け取るデータ項目を定義し、親フローから値を渡します。

▼親フローの作成例



出典：[子フローの作成 - Power Automate | Microsoft Learn](#)

1.2.4. 出力パラメータの返却

子フローの処理結果を出力として返し、親フローで受け取って後続処理に利用します。

▼データを親フローに戻す設定例



1.2.5. 再利用性と保守性の向上

子フローを共通処理として設計することで、複数のフローで再利用でき、変更時の影響範囲を限定できます。

1.3. 外部 API を呼び出してクラウドフローでアクションを実行する

1.3.1. 概要

クラウドフローでは「HTTP」アクションを使用して、外部の WebAPI と通信し、データの取得や更新などの操作を行うことができます。

1.3.2. HTTP アクションの基本構成

以下の基本構成を設定してリクエストを送信します。

- ①メソッド：GET、POST、PUT、DELETE など
- ②URL：API のエンドポイント
- ③ヘッダー：認証情報やコンテンツタイプなど
- ④ボディ：POST や PUT で送信するデータ

▼HTTP コネクタ

HTTP

Parameters Settings Code View About

URI *
Enter request url

Method *
DELETE
GET
PATCH
POST
PUT

[Enter custom value](#)

Cookie
Enter HTTP cookie

Advanced parameters
Showing 0 of 1 Show all Clear all

出典：[導入 - Training | Microsoft Learn](#)

1.3.3. 認証の設定

HTTP コネクタの認証オプションには、None、基本、クライアント証明書、Active Directory OAuth、未加工、マネージド ID があります。
API の仕様に応じた認証方式を設定する必要があります。

▼HTTP コネクタの認証オプション

The screenshot shows the 'HTTP' connector configuration window. The 'パラメーター' (Parameters) tab is selected. The 'URI' field is labeled 'Enter request url'. The 'Method' field is a dropdown menu. The 'Headers' and 'Queries' sections each have a 'キーを入力します' (Enter key) and a '値を入力します' (Enter value) field, along with a plus icon. The 'Body' field is labeled 'Enter request content'. The 'Cookie' field is labeled 'Enter HTTP cookie'. A yellow box highlights the '認証オプション' (Authentication Options) section, which includes a list of options: 'None', '基本' (Basic), 'クライアント証明書' (Client Certificate), 'Active Directory OAuth', '未加工' (Unprocessed), and 'マネージド ID' (Managed ID). Below this list is a dropdown menu currently set to 'None'. There are also buttons for 'すべてクリア' (Clear All) and a close button 'X'.

1.3.4. レスポンスの処理

多くの API は JSON 形式でレスポンスを返します。
JSON 形式で返されるデータを「JSON の解析」アクションを使って構造化データとし、後続のアクションで活用できます。

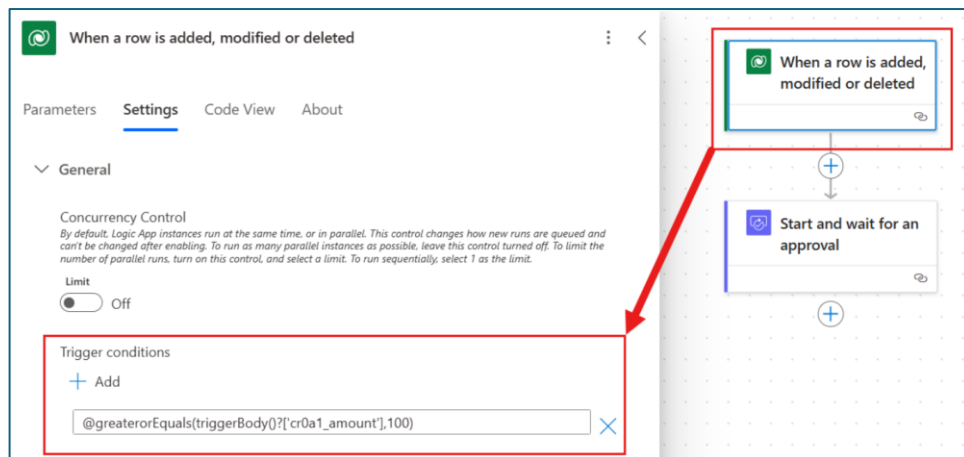
1.4. クラウドフローにトリガー条件とコンカレンシーを実装する

1.4.1. 概要

クラウドフローでは、トリガー条件を設定することで、特定の条件を満たしたときだけフローが実行されるようにできます。これにより、不要な処理を防ぎ効率的な動作が可能になります。
また、コンカレンシー（同時実行）制御を使うことで、複数の処理が同時に走る際の動作を調整できます。これにより、システムの負荷を抑えたり、処理の順序を守ったりすることができ、安定した運用につながります。

1.4.2. トリガー条件の設定方法

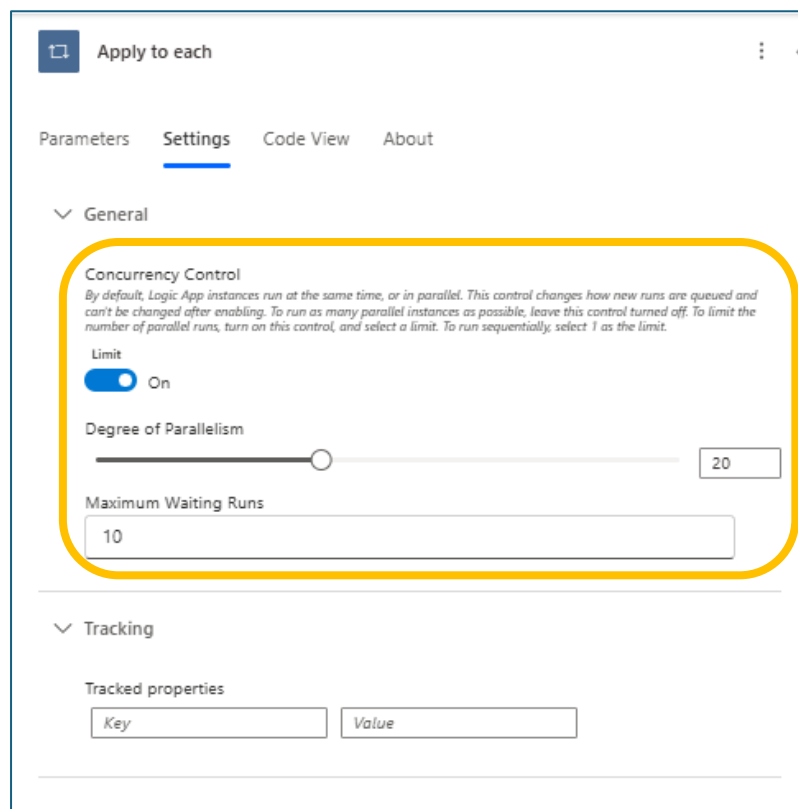
トリガーの「設定」タブ「トリガーの条件」フィールドに式を記述して、実行を制限します。



出典：[Power Automate トリガーを最適化する - Power Automate | Microsoft Learn](#)

1.4.3. コンカレンシー制御の設定

アクションの「設定」タブ「同時実行管理」オプションを有効にし、同時実行数などを設定できます。



出典：[並列実行と同時実行の実装 - Power Automate | Microsoft Learn](#)

1.5. クラウドフローでタイムアウトと再試行のポリシーを実装する

1.5.1. 概要

クラウドフローでは、アクションが失敗した場合に自動で再試行する「再試行ポリシー」や、処理時間が長すぎる場合に中断する「タイムアウト設定」を使うことで、フローの安定性と信頼性を高めることができます

1.5.2. 再試行ポリシーの設定

アクションの「設定」から再試行回数、間隔、指数バックオフ（間隔を徐々に増やす）などを指定できます。デフォルトでは最大4回の再試行が設定されています。

▼再試行ポリシー設定画面

The screenshot shows the '設定' (Settings) tab for the 'Requesting Approval' action. The '再試行ポリシー' (Retry Policy) section is highlighted with a yellow box. It includes a dropdown menu currently set to '既定' (Default), with a list of options: '既定', 'なし' (None), '指数の間隔' (Exponential interval), and '固定間隔' (Fixed interval). The text above the dropdown explains that the retry policy applies to HTTP status codes 408, 429, and 5xx, and that the default is an exponential interval policy based on the performance profile.

1.5.3. タイムアウト（最大実行時間）の設定

アクションのタイムアウト（最大実行時間）▼を指定し、処理が長引いた場合に自動的に中断することが可能です。

The screenshot shows the '設定' (Settings) tab for the 'Requesting Approval' action. The 'アクションのタイムアウト' (Action Timeout) section is highlighted with a yellow box. It includes a text input field where 'PT3H' (3 hours) has been entered. The text above the field indicates that the timeout should be specified in ISO-8601 format.

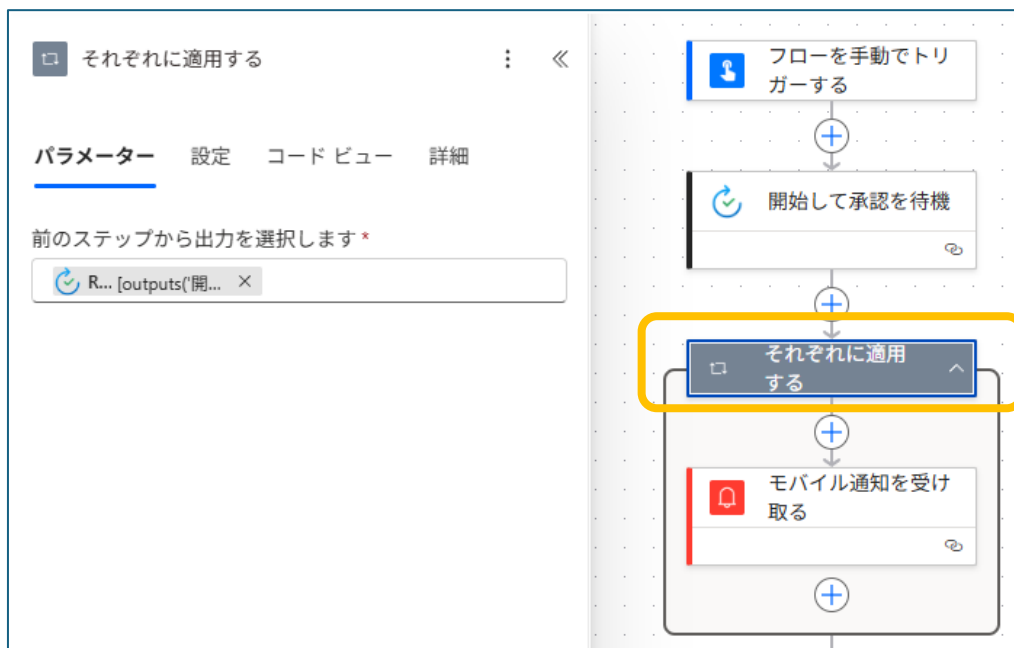
1.6. クラウドフローにデータ オブジェクトとデータ操作を実装する

1.6.1. 概要

クラウドフローでは、JSON 形式のデータを操作することで、複雑な構造のデータを柔軟に扱うことができます。これにより、外部サービスとの連携やデータ整形が容易になります。

1.6.2. 配列の操作

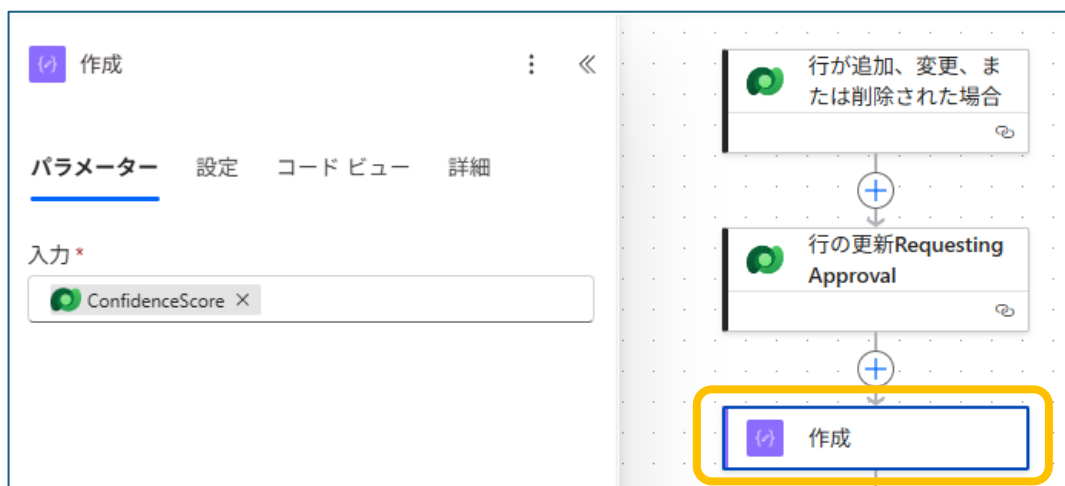
「Apply to each（それぞれに適用する）」アクション▼を使って、配列内の各要素に対して処理を実行する。



1.6.3. オブジェクトの操作

「作成 (Compose)」や「JSON の解析 (Parse JSON)」アクションを使って、オブジェクトの中身を抽出・加工する。

▼「作成」アクション使用例



1.7. クラウドフローで JSON、XML、CSV を含むテキスト解析を実行する

1.7.1. 概要

クラウドフローでは、外部サービスから取得した JSON、XML、CSV などのテキストデータを解析し、必要な情報を抽出してフロー内で活用できます。

1.7.2. JSON の解析

「JSON の解析」アクション▼を使って、JSON 文字列を構造化データとして扱います。

1.7.3. XML の解析

「XPath」関数を使って、XML データから特定の要素を抽出します。

・使用例

①XML データを取得

例：HTTP アクションで外部 API から XML レスポンスを受け取る

②「式」フィールドで XPath 関数を使用

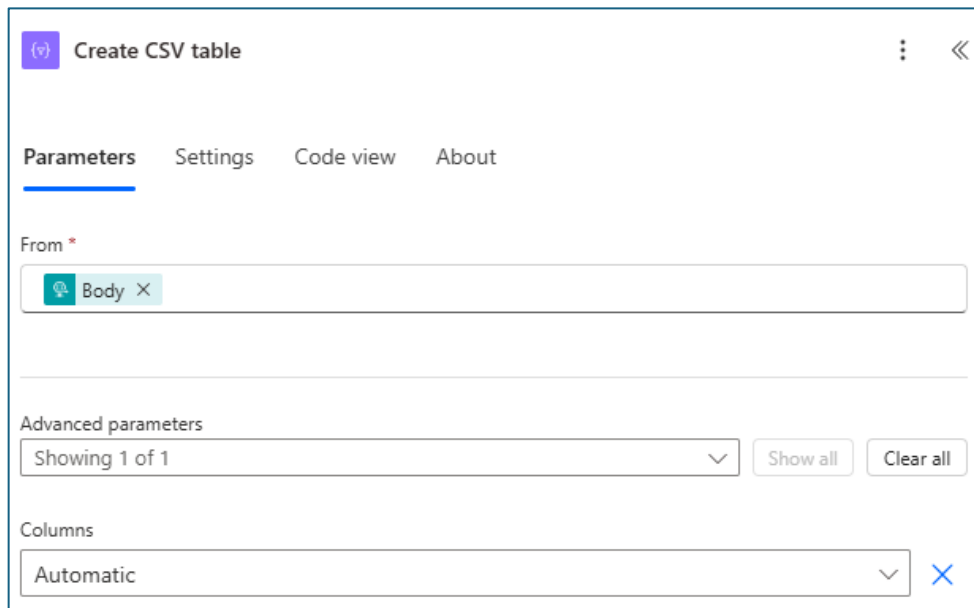
`xpath(xml(outputs('HTTP')), '//*[@customer/name/text()]')`

③抽出結果を後続アクションで使用する

1.7.4. CSV の解析

「CSV テーブルの作成」や「選択」アクションを使って、CSV 形式のデータを生成・加工します。

▼ 「CSV テーブルの作成」アクション



The screenshot shows the 'Create CSV table' action configuration window. At the top, there's a title bar with a purple icon and the text 'Create CSV table'. Below the title bar are four tabs: 'Parameters' (selected), 'Settings', 'Code view', and 'About'. The 'Parameters' tab contains a 'From' section with a dropdown menu showing 'Body' and a close button 'X'. Below this is an 'Advanced parameters' section with a dropdown menu showing 'Showing 1 of 1', a 'Show all' button, and a 'Clear all' button. At the bottom is a 'Columns' section with a dropdown menu showing 'Automatic' and a close button 'X'.

出典：[Power Automate でデータ操作を使用する - Power Automate | Microsoft Learn](#)

2. デスクトップフローを開発する

2.1. UI オプションを実装する

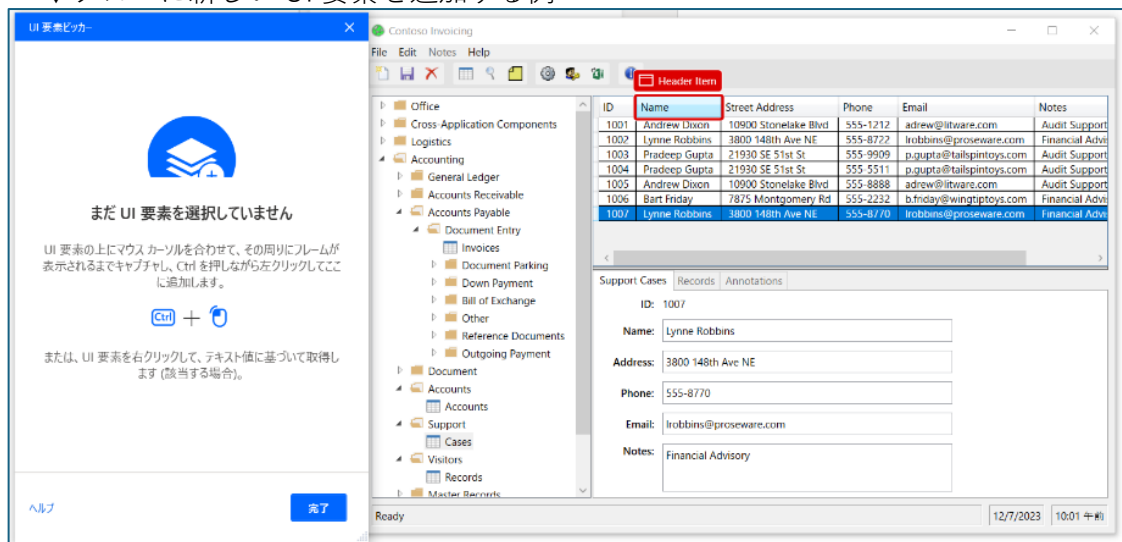
2.1.1. 概要

デスクトップフローでは、対象アプリケーションの UI 要素を操作するために、UI 要素認識・画像認識・キーボード/マウス操作など複数の方法が用意されています。

2.1.2. UI 要素の認識

ウィンドウやボタンなどの UI 要素を識別し、クリックや入力を自動化します。
UIA/MSAA セレクターを使って要素を正確に指定できます。

▼フローに新しい UI 要素を追加する例



出典：[UI 要素を使用して自動化する - Power Automate | Microsoft Learn](#)

2.1.3. 画像認識の利用

UI 要素が認識できない場合、画面上の画像を使って操作対象を特定します。
解像度や DPI が一致していないと認識精度が下がるため注意が必要です。

▼「画面の解像度を設定」アクション設定例



出典：[画面解像度が異なるマシンで画像認識を使用する - Power Automate | Microsoft Learn](#)

2.1.4. キーストロークとマウス操作

キーボード入力やマウスの動きを記録・再生して操作を自動化します。
特殊キーや座標指定も可能で、柔軟な操作ができます。

レコーダーの記録例▶



出典：[デスクトップフローを記録する - Power Automate | Microsoft Learn](#)

2.2. デスクトップフローにデータテーブル、リスト、カスタムオブジェクトを実装する

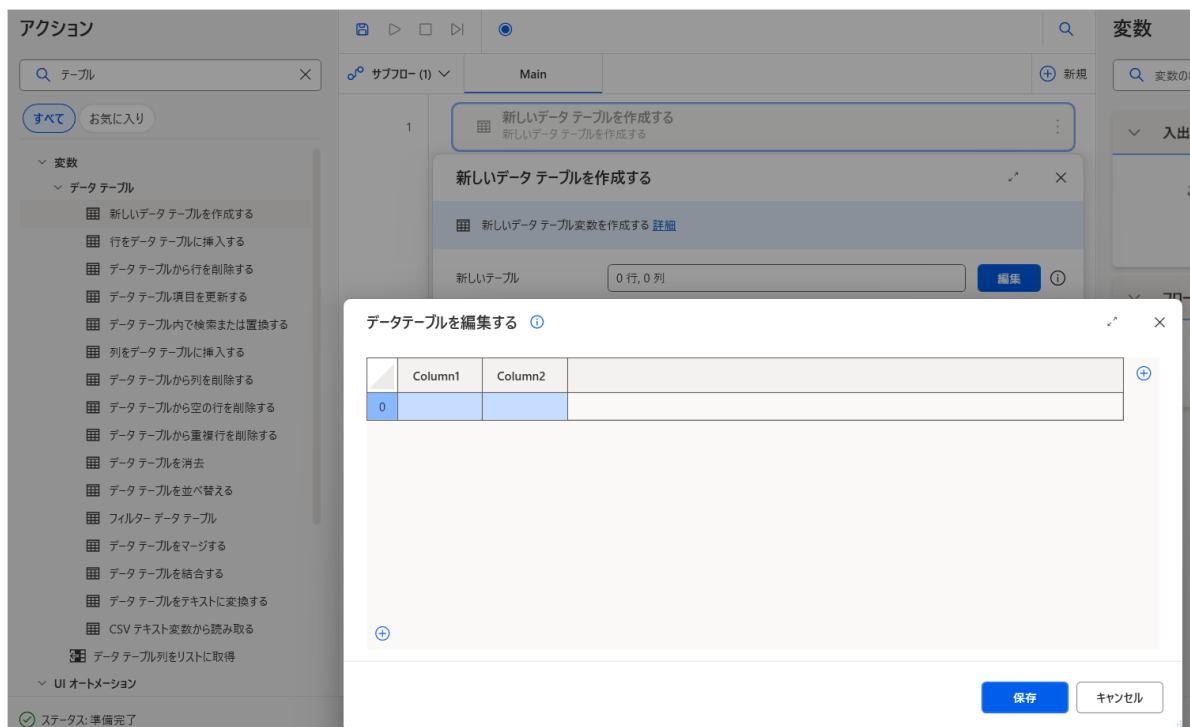
2.2.1. 概要

データ構造を活用することで、複数の値を効率的に管理し、処理を自動化できます。
リストやデータテーブル、カスタムオブジェクトを使うと、複雑なデータ操作が容易になります。

2.2.2. データテーブルの使用

データテーブルは行と列で構成された表形式のデータを扱います。
繰り返し処理や条件分岐で、複数のレコードを効率的に処理できます。

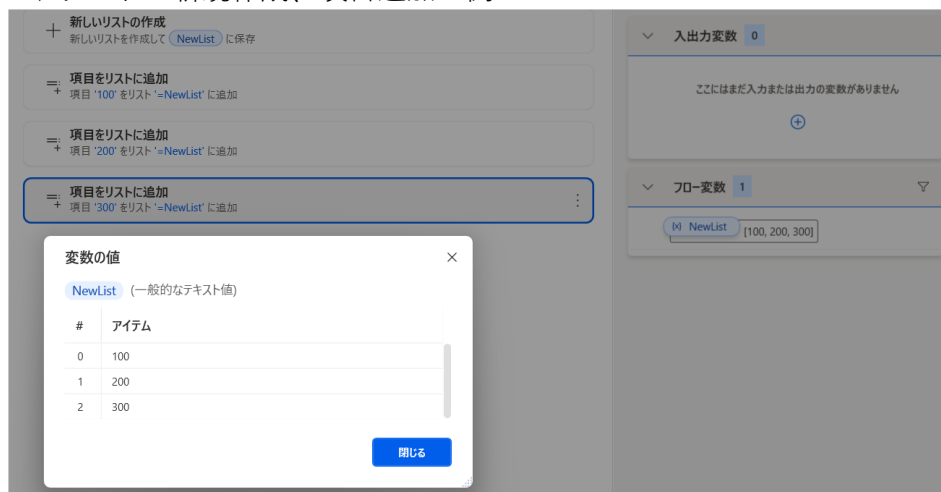
▼「新しいデータテーブルを作成するアクション」



2.2.3. リストの使用

リストは単一列のデータを順番に処理するために使用します。
要素の追加・削除や並び替えが簡単にできます。

▼リストの新規作成、項目追加の例



2.2.4. カスタムオブジェクトの活用

カスタムオブジェクトは複数のプロパティを持つデータ構造です。
柔軟なデータ管理が可能で、複雑な情報をまとめて扱えます。

▼「変数を設定する」アクションを使用して作成する



変数の設定

{x} 新規や既存の変数に値を設定する、新しい変数を作成する、または以前作成した変数を上書きする [詳細](#)

変数: Product {x}

値: `{"Name": 'OLED TV', 'Code': '2544XEY', 'InStock': 'Yes' }` {x} ⓘ

保存 キャンセル

出典：[変数のデータ型 - Power Automate | Microsoft Learn](#)

2.3. デスクトップフローにサブフローを実装する

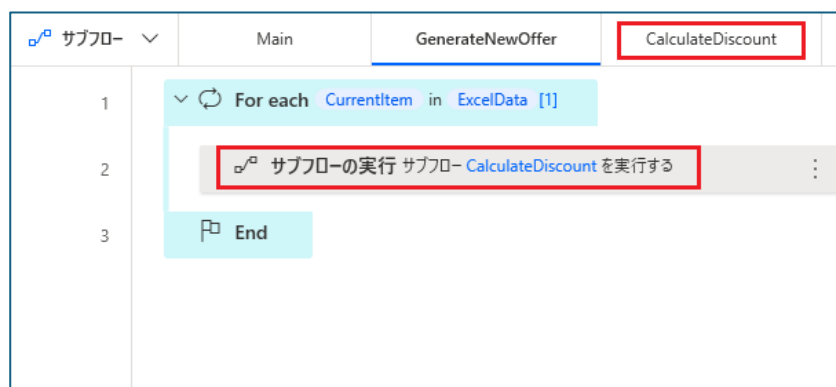
2.3.1. 概要

サブフローは処理を分割し、再利用できるようにする機能です。
これにより、フローの構造を整理しやすくなり、保守性が向上します。

2.3.2. サブフローの作成

メインフロー内で「サブフローの実行」アクションを使って、別の処理を実行します。
複雑な処理を分けることで、フロー全体を見やすくできます。

▼「サブフローの実行」アクション使用例



サブフロー ▼ Main GenerateNewOffer CalculateDiscount

1 For each CurrentItem in ExcelData [1]

2 サブフローの実行 サブフロー CalculateDiscount を実行する

3 End

出典：[フロー制御アクション リファレンス - Power Automate | Microsoft Learn](#)

2.3.3. 入出力の設定

サブフローに渡すパラメータ(入力変数)と戻り値(出力変数)を定義して、データのやり取りを行います。
これにより、サブフローを柔軟に再利用できます。

2.4. 外部 API を呼び出してデスクトップフローでアクションを実行する

2.4.1. 概要

デスクトップフローでは、HTTP リクエストを使って外部サービスと連携できます。API を通じてデータ取得や操作を自動化できます。

2.4.2. Web サービス呼び出しアクション

「Web サービスを呼び出します」アクションで、URL、メソッド（GET/POST など）、ヘッダー、ボディを設定します。認証情報やパラメータもここで指定できます。▼

出典：[HTTP を使用したアクションについての参考情報 - Power Automate | Microsoft Learn](#)

2.4.3. レスポンスの処理

API から取得したレスポンスは変数（WebServiceResponse）に格納され、後続の処理に活用できます。JSON や XML の解析を組み合わせると、必要な情報を抽出できます。

2.5. デスクトップフローにタイムアウトと再試行を実装する

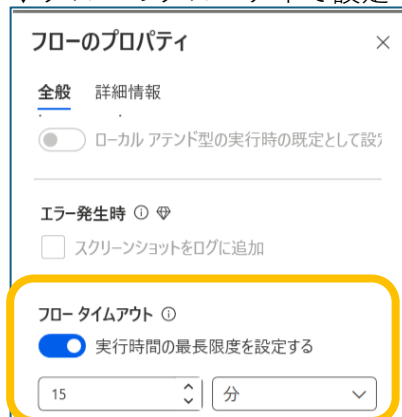
2.5.1. 概要

処理が失敗した場合に再試行を行ったり、一定時間で処理を中断することで、安定した自動化が可能です。

2.5.2. タイムアウトの設定

フロー全体に最大実行時間を指定します。これにより、無限待機や予期しない停止を防ぎます。利用には Premium ライセンスが必要です。

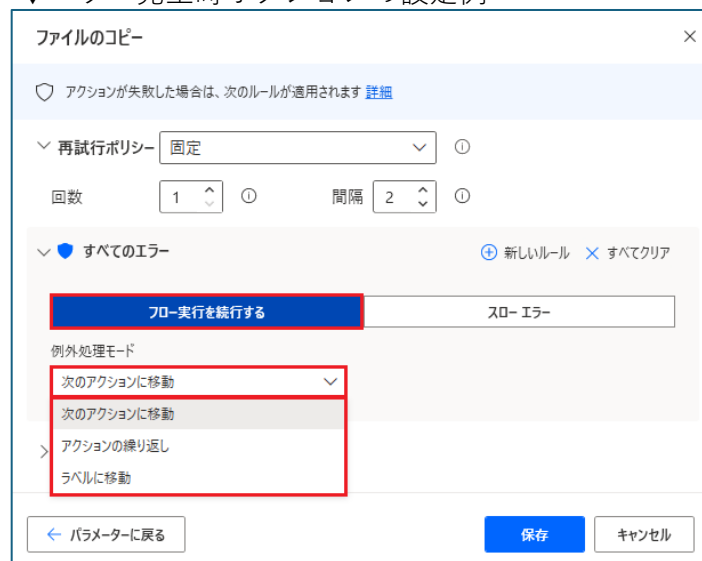
▼フローのプロパティで設定



2.5.3. 再試行の設定

アクションの失敗時に「エラー発生時」オプションで再試行を設定できます。再試行回数や遅延時間を指定し、一時的なエラーに対応します。

▼エラー発生時オプションの設定例



出典：[デスクトップ フローでエラーを処理する - Power Automate | Microsoft Learn](#)

2.6. デスクトップフローにデータオブジェクトとデータ操作を実装する

2.6.1. 概要

デスクトップフローでは、処理の中でデータを保持・加工するために「データオブジェクト」を使用します。単一の値から構造化データまで、目的に応じた型を選択して扱います。

「データ操作」は、変数やテーブル、リストなどのデータを加工・抽出・変換する処理全般を指します。JSON や CSV などの構造化テキストの解析もこの範囲に含まれます。

2.6.2. データオブジェクトの種類

デスクトップフローでは、処理対象に応じて異なるデータ型を使い分けます。

代表的な型はテキスト、数値、ブール値、リスト、データテーブル、カスタムオブジェクトです。

リストは順序付きデータの集まり、データテーブルは行列形式の表構造を持ちます。

カスタムオブジェクトは JSON に似た柔軟な構造を扱う際に有効です。

▼「新しい入力変数」アクションで選択できるデータの種類の

新しい入力変数

入力または出力として使用する新しい変数を追加します [詳細](#)

変数名: NewInput

データの種類: テキスト

既定値: ☒ テキスト

外部名:

説明:

機密情報としてマーク ☐

任意としてマークする ☐

数値

ブール値

カスタム オブジェクト

リスト

データテーブル

インスタンス

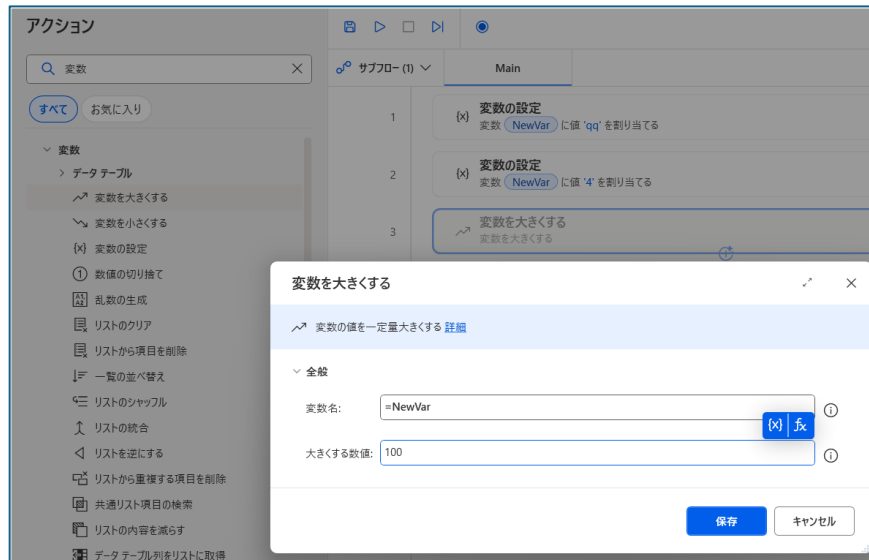
セッション

保存 キャンセル

2.6.3. データ操作の基本

データ操作とは、変数に対して値を設定・変更・抽出する一連の処理を指します。
代表的なアクションには「変数を設定」「変数を大きくする」「リストに項目を追加」「データ テーブルをフィルター処理」などがあります。
数値計算、文字列結合、リストのループ処理などもここに含まれます。
カスタムオブジェクトではキーを指定して値を取得または更新できます。
データの型に合わせた操作アクションを正確に選択することが大切です。

▼「変数を大きくする」アクション例



2.7. デスクトップフローで JSON、XML、CSV などのテキスト解析を実行する

2.7.1. 概要

外部から取得したテキストデータ（JSON、XML、CSV）を解析し、必要な情報を抽出して処理に活用します。データ形式に応じて、専用アクションやスクリプトを使用します。

- ・ JSON：専用アクションなし。スクリプトや文字列操作で対応。
- ・ XML：専用アクションあり。（読み取り、XPath、属性操作）
- ・ CSV：専用アクションあり。（CSV ファイル読み取り）

2.7.2. JSON の解析

API レスポンスなどの JSON 文字列は、文字列操作アクションやスクリプト実行（PowerShell、Python など）で解析します。キーや値を抽出し、変数に格納して後続処理に利用します。

▼「Python スクリプトとの実行」アクション例

Python スクリプトの実行

Python 2 スクリプト コードを実行し、出力を取得します [詳細](#)

パラメーターの選択

全般

実行する Python スクリプト:

```
1 variableName = %Value%
2 print variableName
```

> 詳細

> 生成された変数 [PythonScriptOutput](#) [ScriptError](#)

[エラー発生時](#) [保存](#) [キャンセル](#)

出典：[スクリプトアクションのリファレンス - Power Automate | Microsoft Learn](#)

2.7.3. XML の解析

「ファイルから XML を読み取る」アクションで XM ファイルを読み込み、「XPath 式を実行します」アクションで特定のノードや属性を抽出します。要素や属性の取得・設定・削除も可能です。

▼「XPath 式を実行します」アクション例

XPath 式を実行します

</> 指定された XPath クエリに基づいて、XML ドキュメントから値を抽出します [詳細](#)

パラメーターの選択

全般

解析する XML ドキュメント: %XmlDocument%

XPath クエリ: clientlist[1]/client[1]/email[1]

最初の値のみ取得します: ☒

> 生成された変数 [XPathResults](#)

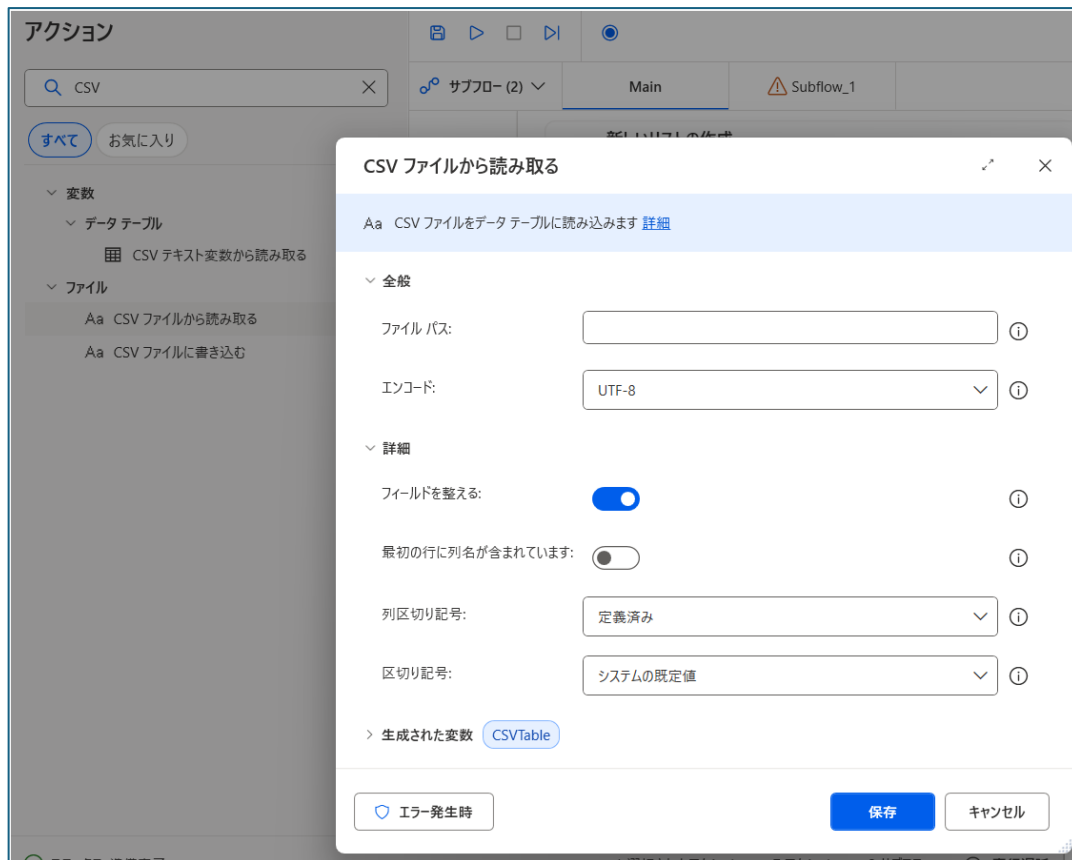
[エラー発生時](#) [保存](#) [キャンセル](#)

出典：[XML を使用したアクションについての参考情報 - Power Automate | Microsoft Learn](#)

2.7.4. CSV の解析

「CSV ファイルから読み取る」アクションで CSV を行と列に分割し、データテーブル(変数 CSVTable に格納)として扱います。カスタム区切り文字やヘッダー行の設定も可能です。

▼「CSV ファイルから読み取る」アクション例



2.8. デスクトップフローにカスタムアクションを実装する

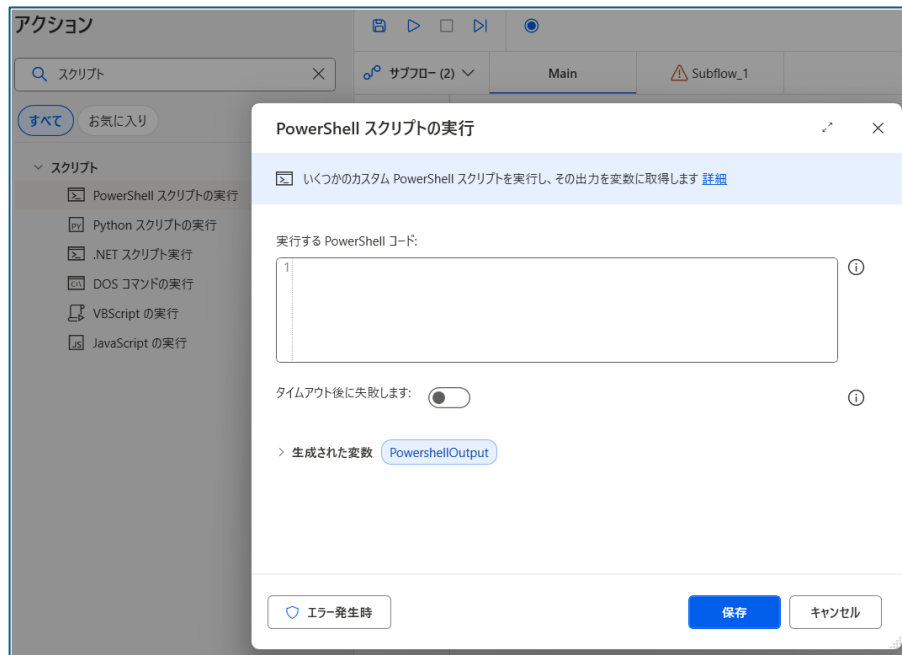
2.8.1. 概要

標準アクションで対応できない処理は、スクリプトや外部ツールを使って実装できます。これにより、より柔軟な自動化が可能になります。

2.8.2. スクリプトの実行

スクリプトアクションを使って、PowerShell や VBScript などを実行します。外部ライブラリや OS コマンドを利用して高度な処理を追加できます。

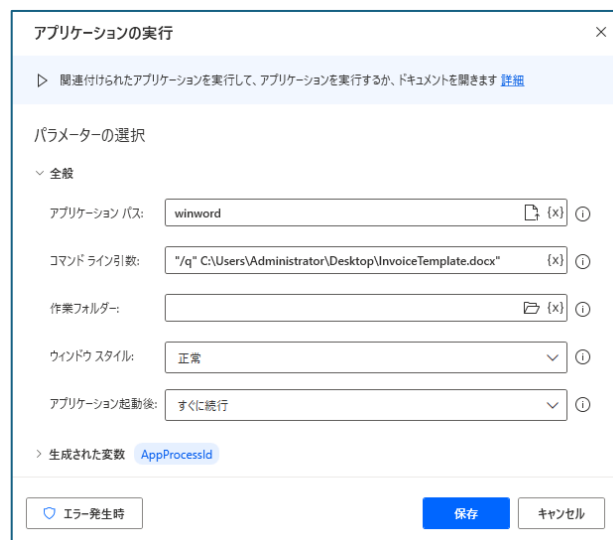
▼「PowerShell スクリプトの実行」アクション例



2.8.3. 外部アプリとの連携

「コマンドラインを実行」アクションや「アプリケーションの実行」アクションで外部アプリを操作します。バッチファイルや CLI ツールを呼び出して、他システムと連携できます。

アプリケーションの実行▶



出典：[システム アクション リファレンス - Power Automate | Microsoft Learn](#)

3. クラウドとデスクトップのフローにロジックを実装する

3.1. ループを含むクラウドとデスクトップのフローにフロー制御を実装する

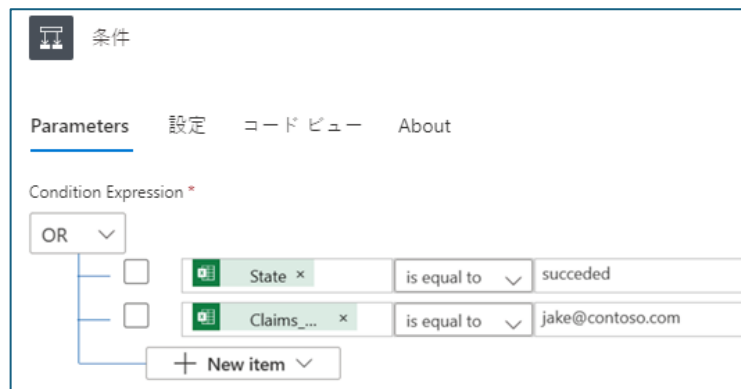
3.1.1. 概要

条件分岐や繰り返し処理を使って、フローの実行順序や回数を制御します。

3.1.2. 条件分岐の実装

「条件」アクションを使って、条件に応じた処理を分岐させます。

▼クラウドフローの「条件」アクション例



出典：[Power Automate で条件式を使用する - Power Automate | Microsoft Learn](#)

3.1.3. ループ処理の実装

「Apply to each」や「Do until」などを使って、配列や条件に基づく繰り返し処理を行います。

▼デスクトップフローの「ループ条件」アクション例



出典：[ループの使用 - Power Automate | Microsoft Learn](#)

3.2. クラウドフローに式を実装する

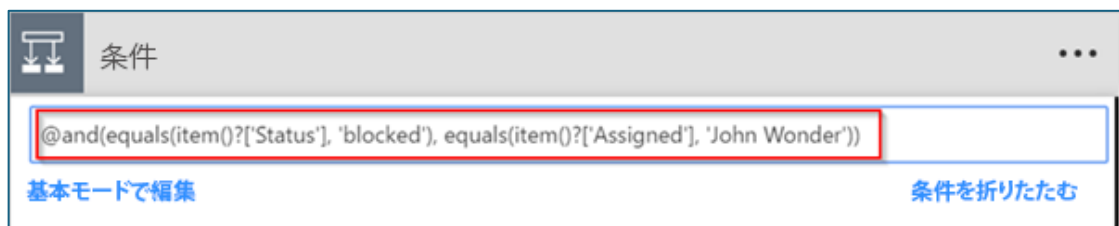
3.2.1. 概要

クラウドフローでは「式」を使うことで、データの加工や条件判定を柔軟に行うことができます。たとえば、文字列の結合、数値の計算、条件分岐などが可能です。

3.2.2. 式の記述方法

クラウドフローの式は、Power Fx に似た構文で記述されます。関数や演算子を組み合わせて、動的な値の取得や条件判定を行います。式は「動的なコンテンツ」や「式エディター」から挿入できます。

▼and 式の例



出典：[Power Automate で条件式を使用する](#) - Power Automate | Microsoft Learn

3.2.3. よく使う関数

クラウドフローでよく使われる関数には以下のようなものがあります。

- ・ if() : 条件に応じて値を返す
- ・ concat() : 文字列を結合する
- ・ length() : 文字列や配列の長さを取得する
- ・ contains() : 指定の値が含まれているか判定する

3.3. クラウドとデスクトップのフローの変数アクションを実装する

3.3.1. 概要

クラウドフローやデスクトップフローでは、変数を使ってデータの一時保存や加工が可能です。変数はフロー内での状態管理や、複数のアクション間でのデータの受け渡しに活用されます。

3.3.2. 変数の作成と初期化

クラウドフローでは「変数を初期化する」アクションを使って、変数の型（文字列、数値、ブール値など）と初期値を設定します。

▼「変数を初期化する」アクション例



The screenshot shows the 'Initialize variable' action interface. It has tabs for 'Parameters', 'Settings', 'Code view', and 'About'. Under 'Parameters', the 'Name' field contains 'Count', the 'Type' dropdown is set to 'Integer', and the 'Value' field contains '0'.

出典：[Power Automate の変数に値を保存して管理する - Power Automate | Microsoft Learn](#)

デスクトップフローでは、変数ペインから入力変数を作成し、型や既定値を指定します。

▼変数ペイン



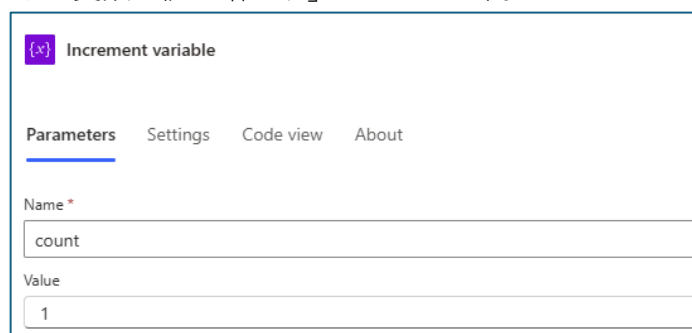
The screenshot shows the 'Variables' pane. It has a search bar labeled '変数の検索'. Below it, there are two sections: '入出力変数 1' (Input/Output variables) containing 'ClipboardOutput', and 'フロー変数 3' (Flow variables) containing 'CurrentDateTime...' (value: 2023/12/11 9:28:58), 'ExcelData' (value: 10 行, 6 列), and 'ExcelInstance' (value: Excel インスタンス). A red box highlights the bottom of the pane.

出典：[変数と変数ペインを管理する - Power Automate | Microsoft Learn](#)

3.3.3. 値の設定と更新

クラウドフローでは「変数の設定」「変数の値を増やす」などのアクションを使って、変数の値を変更できます。

▼「変数の値を増やす」アクション例




The screenshot shows the 'Increment variable' action interface. It has tabs for 'Parameters', 'Settings', 'Code view', and 'About'. Under 'Parameters', the 'Name' field contains 'count' and the 'Value' field contains '1'.

出典：[Power Automate の変数に値を保存して管理する - Power Automate | Microsoft Learn](#)

デスクトップフローでも、変数の値を設定したり、デバッグ中に編集することが可能です。

▼「変数の設定」アクション例



変数の設定

{x} 新規や既存の変数に値を設定する、新しい変数を作成する、または以前作成した変数を上書きする [詳細](#)

変数: Price {x}

値: %10 + 10 * VAT% {x} ①

保存 キャンセル

出典：変数のデータ型 - Power Automate | Microsoft Learn

3.4. クラウドフローのアクションでセキュリティで保護された入力と出力のデータを実装する

3.4.1. 概要

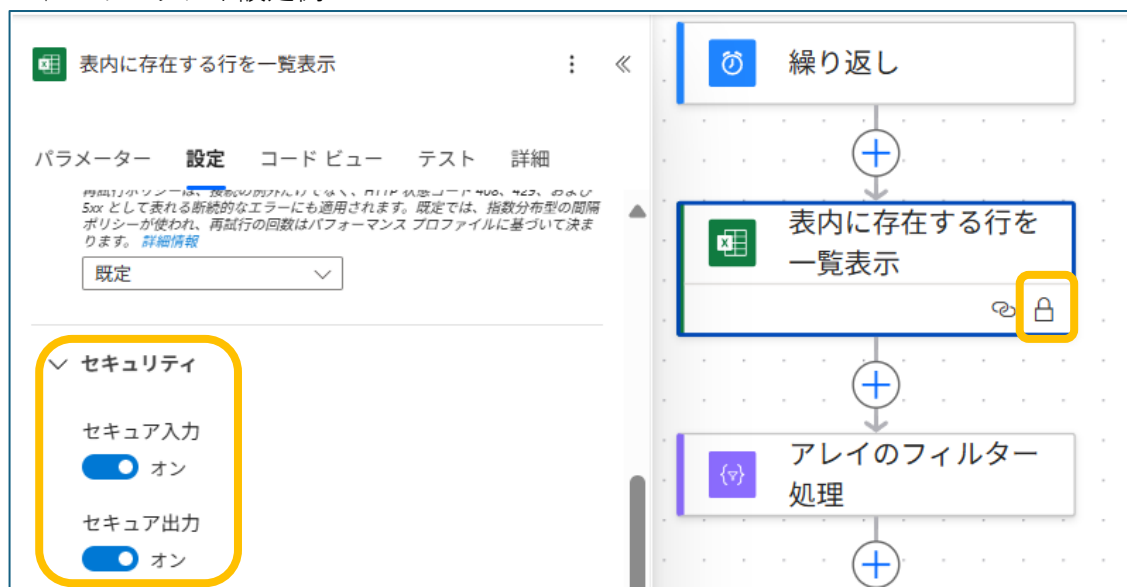
クラウドフローで機密情報（パスワード、個人情報など）を扱う場合、アクションの入力や出力を非表示にすることで、実行履歴などからの情報漏洩を防ぎます。

3.4.2. セキュリティ設定の方法

アクションの「設定」>「セキュリティ」から「セキュア入力」「セキュア出力」を有効にすることで、機密情報がログに表示されなくなります。

アクションにロックアイコンが表示され、保護が有効であることが確認できます。

▼セキュリティ設定例



表内に存在する行を一覧表示

パラメーター 設定 コード ビュー テスト 詳細

既定

✓ セキュリティ

セキュア入力 オン

セキュア出力 オン

繰り返し

表内に存在する行を一覧表示

アレイのフィルター処理

3.5. デスクトップフローでセキュリティで保護された変数を実装する

3.5.1. 概要

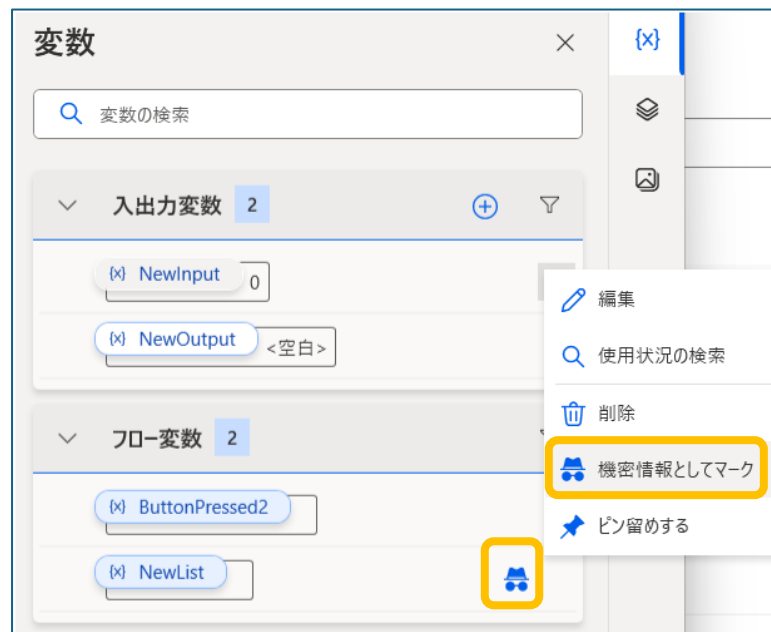
パスワードなどの機密情報を変数として扱う際は、内容を非表示にすることで安全に処理できます。これにより、ログや履歴に機密情報が表示されるのを防ぎます。

3.5.2. セキュリティ変数の作成

デスクトップフローでは、変数ペインで「機密としてマーク」オプションを有効にすることで、セキュリティで保護された変数を作成できます。

これにより、実行中のログや UI に値が表示されなくなります。

▼変数ペインの「機密としてマーク」



3.6. キュー内のデスクトップフローの優先度を実装する

3.6.1. 概要

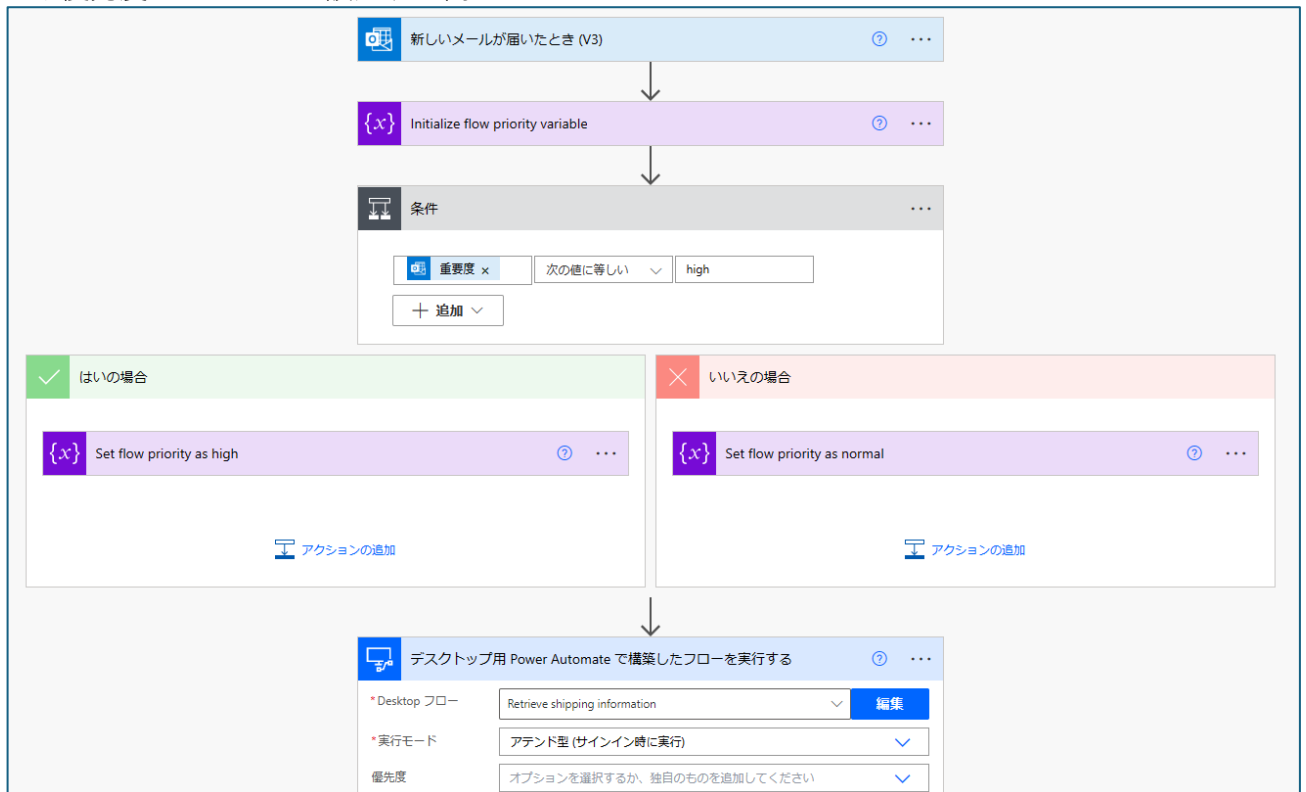
複数のデスクトップフローが同時に実行待ちになる場合、優先度を設定することで重要な処理を先に実行できます。

3.6.2. 優先度の設定方法

デスクトップフローの実行時に「優先度」パラメータを設定することで、キュー内の順序を制御できます。

優先度は「高」「標準」などから選択でき、キューに入れられた時間と組み合わせて実行順が決定されます。

▼優先度パラメータを設定する例



出典：[デスクトップ フロー キューを監視する - Power Automate | Microsoft Learn](#)

3.7. クラウドとデスクトップのフローで例外処理ブロックを実装し、システム例外を処理する

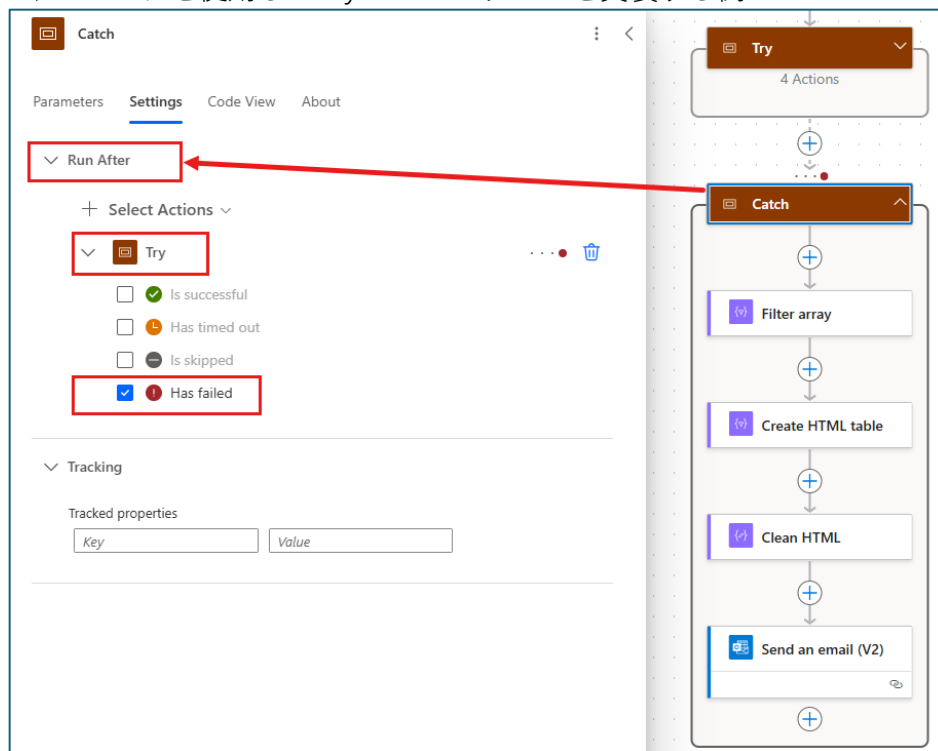
3.7.1. 概要

クラウドフローやデスクトップフローの実行中に発生するエラー（システム例外）に対して、処理を中断せずに対応するための仕組みです。

3.7.2. 例外処理の実装方法

クラウドフローでは「スコープ」アクションを使い、失敗時の処理（try-catch のような構造）を定義します。

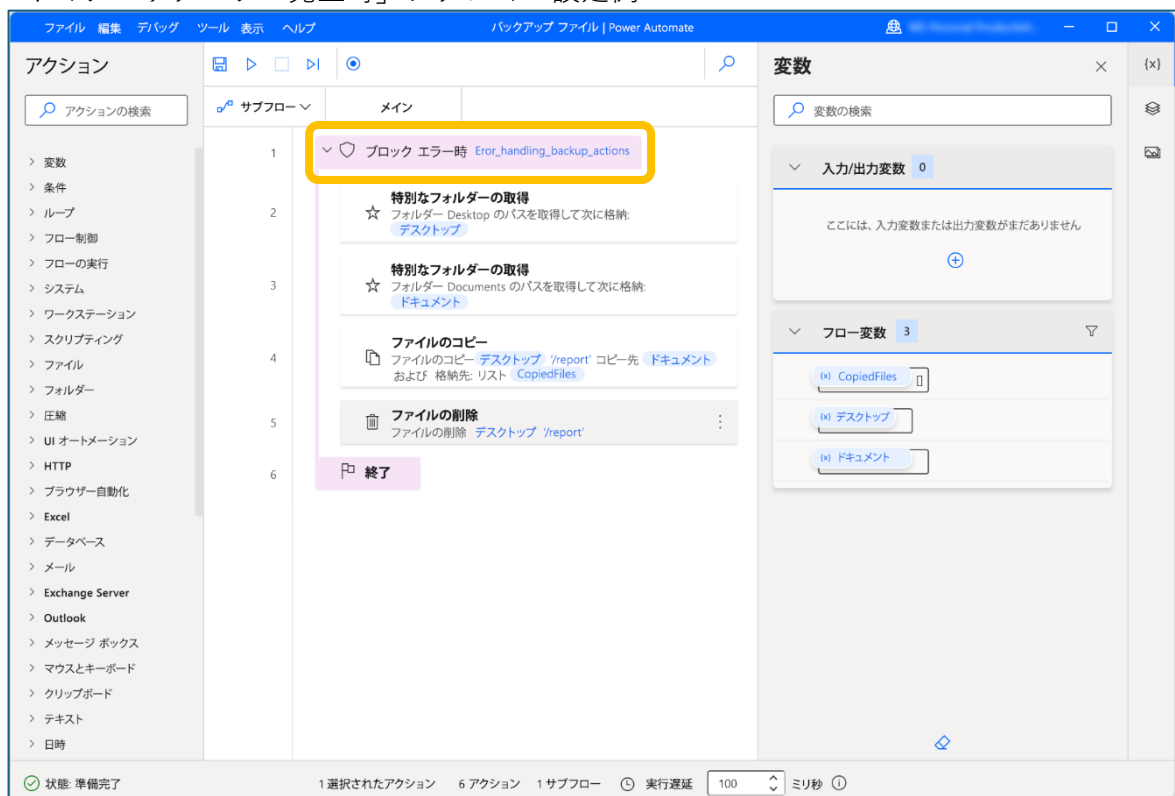
▼スコープを使用して try-catch パターンを実装する例



出典：堅牢なエラー処理を採用する - Power Automate | Microsoft Learn

デスクトップフローでは、各アクションやアクションブロックに対して「ブロックエラー発生時」アクションを設定し、再試行や代替処理を行うことができます。

▼「ブロックエラー発生時」アクション設定例



出典：デスクトップ フローでエラーを処理する - Power Automate | Microsoft Learn

3.8. クラウドとデスクトップのフローでエラー処理ルーチンを実装し、ビジネス例外を処理する

3.8.1. 概要

業務上の条件に合わない場合（ビジネス例外）には、処理を中断するのではなく、条件に応じて処理を分岐させることで、業務ロジックに沿った対応が可能になります。

3.8.2. 条件分岐による処理

クラウドフローでは「条件」アクションや「スコープ」アクションを使って、データの不備や条件未達を検出し、別の処理に切り替えることができます。

▼「条件」アクションの判断で処理を切り替える例



デスクトップフローでは、アクションの「エラー発生時」設定を使って、業務例外に応じたサブフローの呼び出しや変数の設定が可能です。

▼アクションの「エラー発生時」

出典：[デスクトップ フローでエラーを処理する - Power Automate | Microsoft Learn](#)

3.9. クラウドとデスクトップのフローに作業キューを実装する

3.9.1. 概要

作業キューを使うことで、複数の処理対象（例：請求書、申請データなど）を順番に処理でき、効率的な自動化が可能になります。

キューは非同期処理や優先度制御にも対応しています。

3.9.2. キューの構成

Dataverse などに作業キュー項目を登録し、クラウドフローやデスクトップフローで順次取り出して処理します。

キュー項目には、処理対象のデータ、優先度、状態などの情報を含めることができます。

▼作業キュー項目の例

作業キュー > Demo PAD Queue > 作業キュー項目							
名前	キュー...	完了日	優先度	状態	延滞期限	有効期限	入力
I-1025	12月12日 14:48	—	標準	🕒 キュー登録...	—	12月13日 14:48	📄 ["WorkQueueItemId":"67b2184d-8a...
I-1030	12月12日 14:48	—	標準	🕒 キュー登録...	—	12月13日 14:48	📄 ["WorkQueueItemId":"989ae0b3-a4...
I-1027	12月12日 14:48	—	標準	🕒 キュー登録...	—	12月13日 14:48	📄 ["WorkQueueItemId":"23b08df8-49...
I-1028	12月12日 14:48	—	標準	🕒 キュー登録...	—	12月13日 14:48	📄 ["WorkQueueItemId":"2a3425b7-3e...

出典：[作業キュー項目の処理、追加、更新、およびキューへの再登録 - Power Automate | Microsoft Learn](#)

4. カスタムコネクタを構築し、コネクタの構成を実装する

4.1. カスタムコネクタを構築する

4.1.1. 概要

標準コネクタで対応できない外部サービスと連携するために、独自の API 仕様に基づいたカスタムコネクタを作成します。REST API や SOAP API など、任意のサービスと接続可能です。

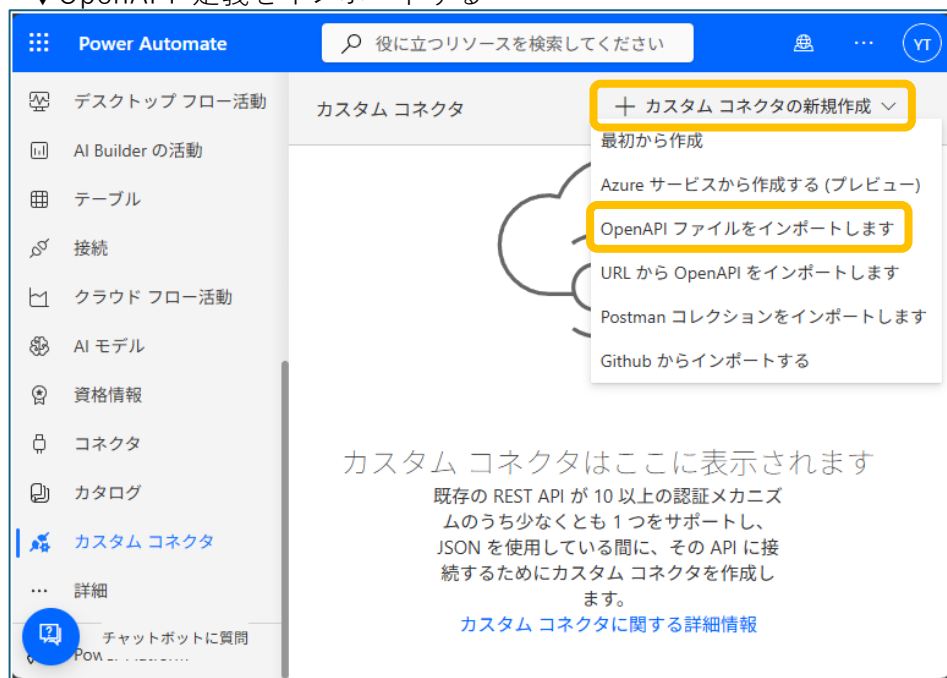
4.1.2. OpenAPI 仕様の利用

API の定義を JSON または YAML 形式で記述し、OpenAPI 2.0 形式でインポートすることで、コネクタの動作を定義します。OpenAPI 3.0 は未対応です。

4.1.3. コネクタの登録方法

Power Automate の「カスタムコネクタ」画面から、OpenAPI 定義ファイルをアップロードして登録します。ウィザード形式で、API のホスト、ベース URL、アクションなどを設定できます。

▼OpenAPI 定義をインポートする



4.2. カスタムコネクタの認証を実装する

4.2.1. 概要

API との安全な通信のために、認証方式を設定します。

これにより、ユーザーが接続時に認証情報を入力し、安全なアクセスが可能になります。

4.2.2. 認証方式の選択

API キー、OAuth2.0、Basic 認証など、API 仕様に応じた方式を選択します。

Microsoft Entra ID（旧 Azure AD）を使った認証も可能です。

▼認証タイプの選択画面



コネクタ名 SampleConnector

1. 全般 > **2. セキュリティ** > 3. 定義 > 4. コード > 5. テスト

Swagger エディター

セキュリティ

認証の種類を選択し、必須フィールドに入力して、カスタム コネクタのセキュリティを設定します。
[詳細情報](#)

認証タイプ

API によって実装される認証の種類を選びます *

- 認証なし
- 認証なし
- ← 基本認証
- API キー
- OAuth 2.0

4.2.3. 認証情報の入力

ユーザーが接続時に必要な認証情報（API キー、クライアント ID など）を入力できるように、ラベルやパラメータの場所（ヘッダー、クエリなど）を設定します。

▼認証情報入力の設定画面



コネクタ名 SampleConnector

1. 全般 > **2. セキュリティ** > 3. 定義 > 4. コード > 5. テスト

Swagger エディター

コネクタの作成

キャンセル

セキュリティ

認証の種類を選択し、必須フィールドに入力して、カスタム コネクタのセキュリティを設定します。
[詳細情報](#)

認証タイプ

API によって実装される認証の種類を選びます *

API キー

API キー

ユーザーは接続の作成時に API キーを入力する必要があります

パラメーターのラベル *

パラメーター名 *

パラメーターの場所 *

ヘッダー

← 全般

定義 →

4.3. カスタムコネクタのポリシーテンプレートを実装する

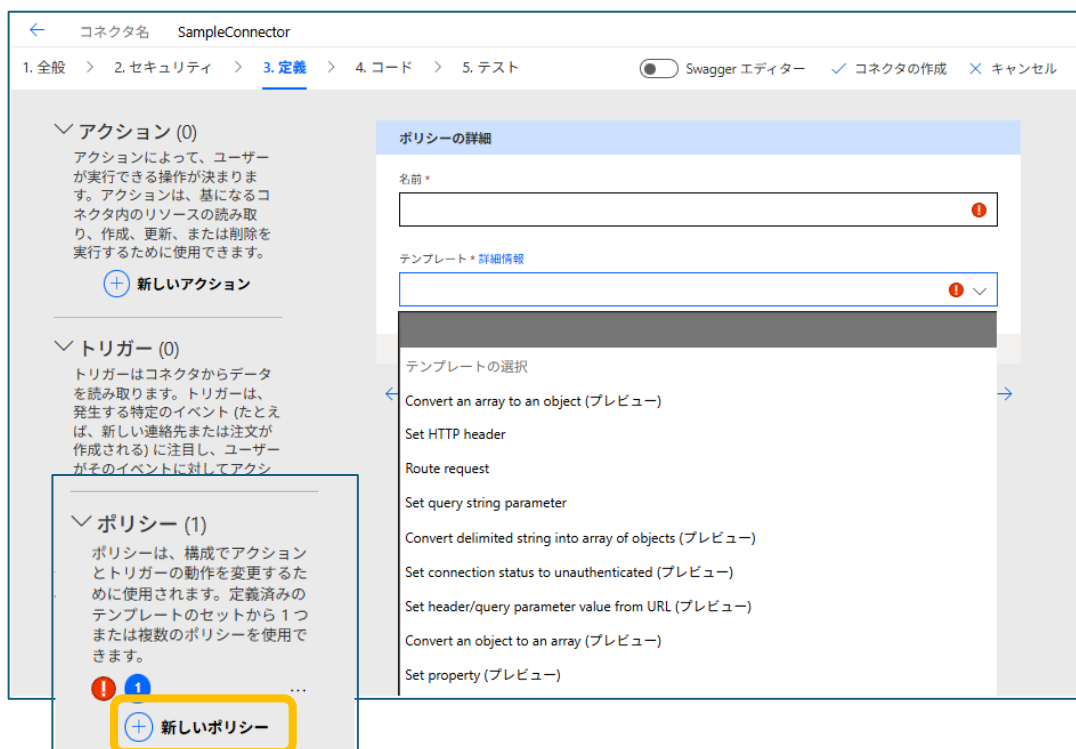
4.3.1. 概要

ポリシーテンプレートを使うことで、コネクタの動作に制限や条件を加えることができます。たとえば、スロットル制限やルーティング制御などが可能です。

4.3.2. 使用例

特定の IP アドレスからのアクセス制限、レスポンスの加工、環境変数の使用などを設定できます。テンプレートはウィザードから追加できます。

▼新しいポリシーの追加画面



4.4. カスタムコネクタでコードを開発する

4.4.1. 概要

コネクタの動作に合わせて、必要なコード（スクリプトや関数）を記述して処理を制御します。これは、リクエストの前処理やレスポンスの整形などに使われます。

4.4.2. 実装例

OpenAPI 拡張機能（x-ms-summary, x-ms-visibility, x-ms-dynamic-values など）を使って、動的な UI 表示やデータ変換をコードで定義します。

- x-ms-summary のコード例

```
"actions" {  
  "Send_an_email": {  
    /// Other action properties here...  
    "parameters": [{  
      /// Other parameters here...  
      "x-ms-summary": "Subject",  
      /// Other parameters here...  
    }]  
  }  
},
```

出典：[OpenAPI 拡張機能からカスタム コネクタを作成する | Microsoft Learn](#)

5. 自動化インフラストラクチャの管理の実行

5.1. 資格情報の管理プラクティスを推奨する

5.1.1. 概要

自動化で使用する資格情報を安全に管理することは、環境全体のセキュリティ維持に欠かせません。パスワードやトークンを平文で保存すると、情報漏えいのリスクが高まります。Power Automate や Dataverse では暗号化された仕組みを利用して保護します。これにより、信頼性の高い自動化基盤を構築できます。

5.1.2. 資格情報の保存方法

「資格情報」ではユーザー資格情報を安全に格納できます。また、セキュリティで保護された「変数」を使用すると、フロー内で値が非表示になります。これにより、他の利用者にパスワードが漏れるリスクを最小化できます。

▼PowerAutomate の資格情報作成画面例

出典：[Azure Key Vault 資格情報を作成する - Power Automate | Microsoft Learn](#)

5.1.3. アクセス制限の設定

資格情報や変数へのアクセスは、環境やロールに応じて制御できます。管理者はデータ損失防止ポリシー(DLP)を利用して利用可能なコネクタ範囲を制限できます。また、共有設定を適切に行うことで不正アクセスを防止します。アクセス権を定期的に見直すことも重要です。

▼DLP 違反

⚠ データ損失防止ポリシー違反

フローは実行されませんでした。組織によって設定されている 1 つ以上のデータ損失防止 (DLP) ポリシーに違反しているためです。ポリシーに違反するアクションを削除または無効にするか、組織の管理者に問い合わせてください。

[詳細](#)

詳細

[Contoso データベース] ポリシーの

- アクション グループ [データベース] は、組織によって業務用とマークされているため、非業務アクション グループ [Excel] と共に使用することはできません

5.2. オンプレミスデータゲートウェイを利用してクラウドフローのリソースを接続する

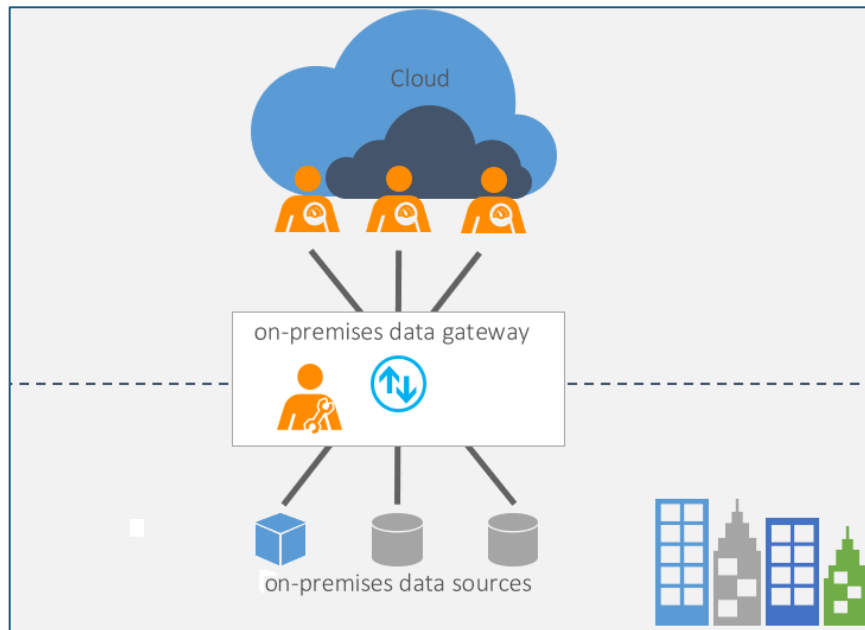
5.2.1. 概要

オンプレミスデータゲートウェイは、クラウドフローから社内ネットワーク上のデータへ安全にアクセスするための橋渡しです。

これにより、クラウド環境とオンプレミスシステムをシームレスに統合できます。社内の SQL Server やファイル共有を外部に公開せずに利用できます。

企業のハイブリッド環境で特に重要な役割を果たします。

▼ゲートウェイのしくみ



出典：[オンプレミス データ ゲートウェイとは | Microsoft Learn](#)

5.2.2. ゲートウェイのインストールと構成

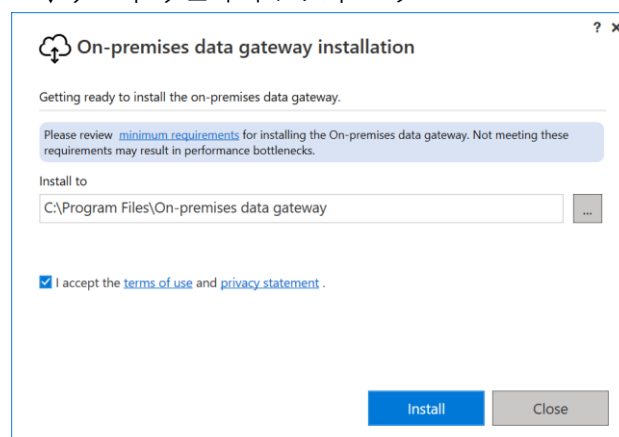
オンプレミスのサーバーや常時稼働マシンにゲートウェイをインストールします。

Power Platform 管理センターから登録し、環境とリンクさせます。

企業ポリシーに応じて標準モードまたはパーソナルモードを選択します。

適切なネットワークポート設定も忘れずに確認します。

▼ゲートウェイインストーラ



出典：[オンプレミス データ ゲートウェイをインストールする | Microsoft Learn](#)

5.2.3. 接続の利用方法

ゲートウェイを利用する接続を作成し、クラウドフローで使します。例として、オンプレミスの SQL Server に接続してデータを取得できます。

接続先はゲートウェイを通じて認証され、通信は暗号化されます。

環境移行時には接続の再構成も必要になる場合があります。

5.3. Microsoft Dataverse ソリューションでコンポーネントを構築する

5.3.1. 概要

Microsoft Dataverse ソリューションは、アプリやフロー、テーブルなどを一元管理する枠組みです。ソリューション単位でコンポーネントをまとめることで、構成変更や配布が容易になります。環境間の移行やバージョン管理にも対応しています。

アプリ開発のライフサイクル全体での統一的管理に役立ちます。

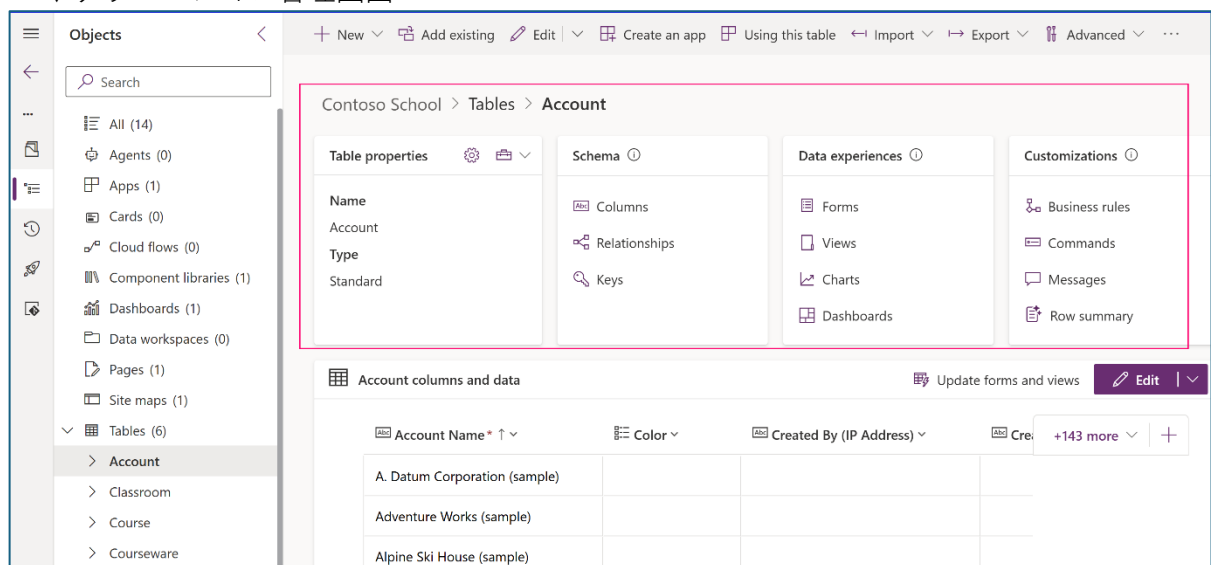
5.3.2. ソリューションの作成

Power Apps や Power Automate の管理画面から新しいソリューションを作成します。

名前空間や発行元を指定し、アプリやフローなどを追加して構成します。必要に応じて環境変数や接続参照を定義して柔軟性を高めます。

ソリューションは再利用性と保守性を高める中心要素です。

▼ソリューション管理画面



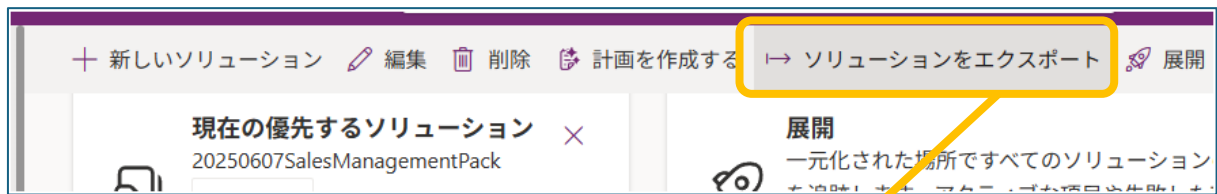
出典：[Power Apps でソリューションを作成する - Power Apps | Microsoft Learn](#)

5.3.3. 環境間の移行

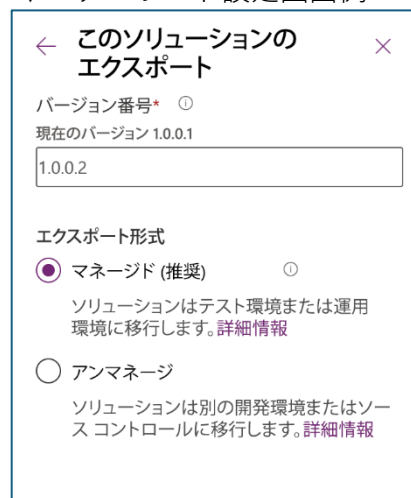
開発環境から本番環境へソリューション単位でエクスポート／インポートします。

依存関係や接続参照を事前に確認しておくことが重要です。

マネージドソリューションを使うと、本番環境での変更を制限できます。これにより、安全で標準化されたリリース運用が可能になります。

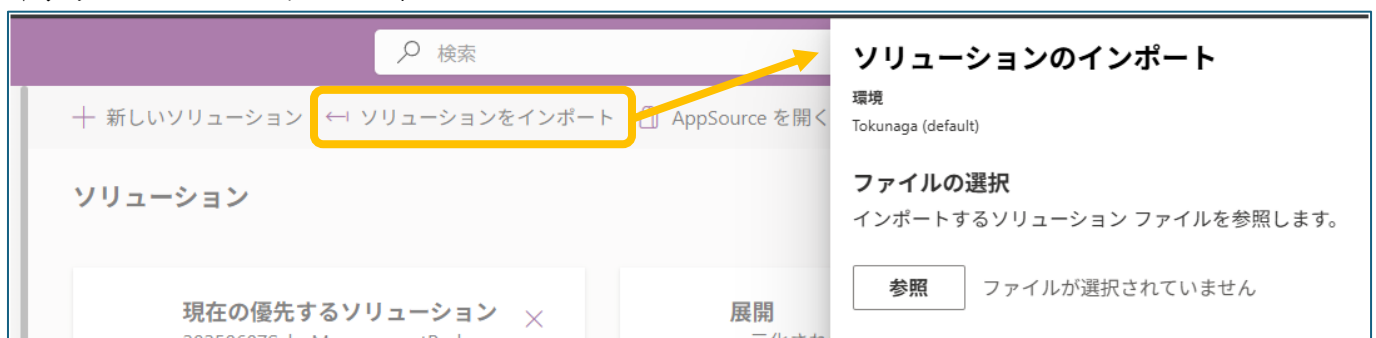


▼エクスポート設定画面例



出典：[ソリューションのエクスポート - Power Apps | Microsoft Learn](#)

▼ソリューションのインポート



6. 自動化をテストし、開発作業を完了する

6.1. クラウドフローをテストする

6.1.1. 概要

クラウドフローをテストすることで、設計通りに動作しているか、条件分岐やデータ取得のロジックが正しく実行されているかを確認します。

テスト工程はエラーの早期発見と修正に役立ち、フローの信頼性を高めます。

また、複雑なビジネスプロセスを自動化する場合は、テストを段階的に実施することが推奨されます。

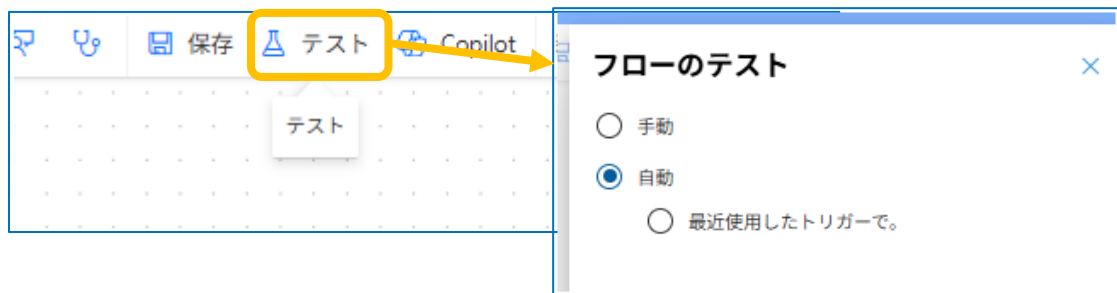
6.1.2. テストの実行方法

Power Automate ポータルの「テスト」ボタンを使用し、手動で実行するか、指定したトリガー条件で検証します。

実データを使う前にテストデータを用いることで、安全に動作を確認できます。

条件分岐やループが多い場合は、複数パターンのテストを行うと効果的です。

▼「テスト」の実行



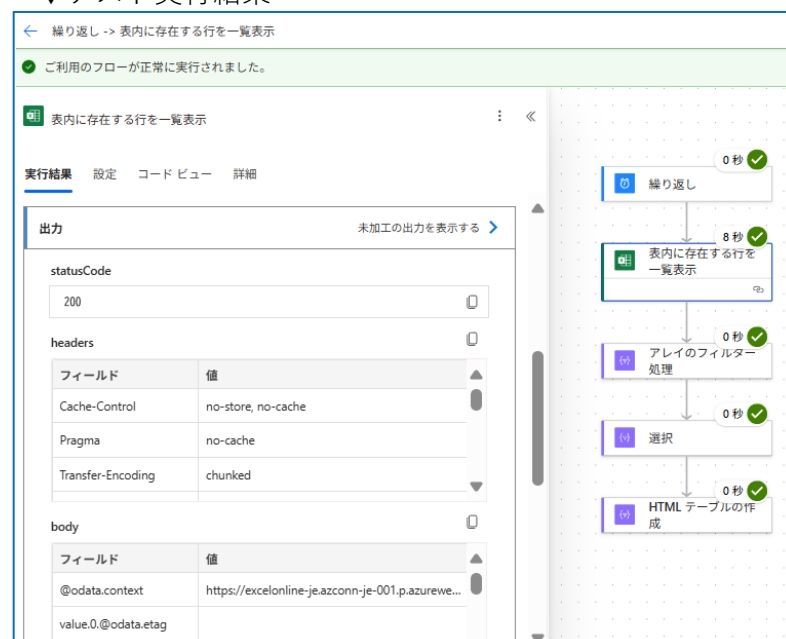
6.1.3. 実行履歴の確認

テスト実行後は、実行履歴から各アクションのステータスや出力内容を確認します。

「成功」「失敗」「スキップ」などの結果を分析し、処理の流れを可視化できます。

エラーが発生した場合は詳細情報から原因を特定し、再実行や修正を行います。

▼テスト実行結果



6.2. デスクトップフローをテストする

6.2.1. 概要

デスクトップフローでは、実際のアプリケーション操作を通じてフローの正確性と安定性を確認します。

手動でステップを実行しながら、UI 操作の認識や変数の動作を確認できます。これにより、実運用環境でのエラーや処理停止を防ぐことができます。

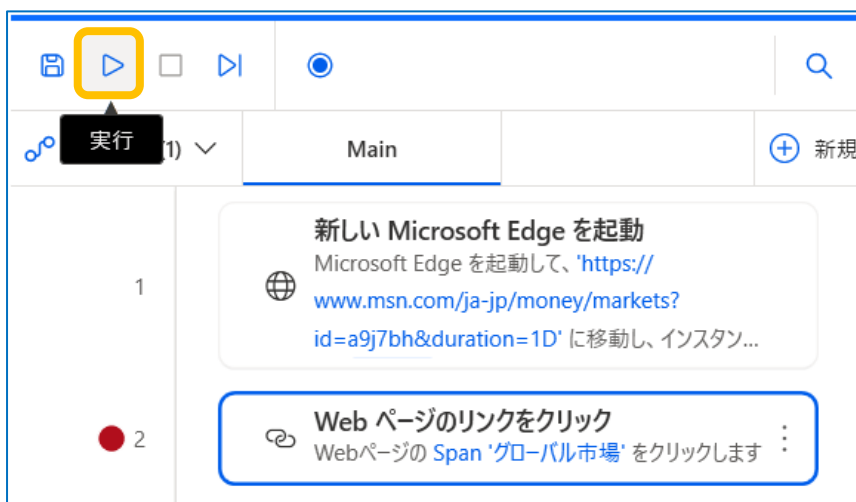
6.2.2. テストの実行方法

デスクトップフローを開き、「実行」ボタンから手動で実行します。

ステップごとに停止・確認しながら動作を追跡できるため、細かい不具合も見つけやすいです。

アクションのパラメータや待機時間を調整しながら安定化を図ります。

▼「実行」ボタン



6.2.3. ログの確認

フロー実行後は、画面下部の「実行ログ」や「詳細ログ」を確認して、各アクションの結果を分析します。

ログには各アクションの成功・失敗の情報や変数値の変化をリアルタイムで確認でき、問題解決の手掛かりになります。

▼実行ログペイン



6.3. 環境変数と構成ファイルを利用して構成を管理する

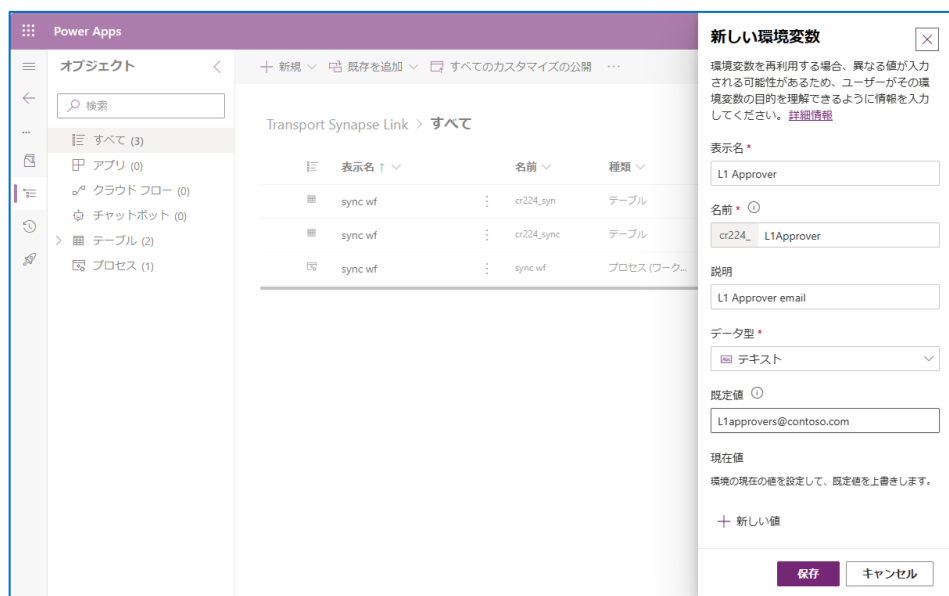
6.3.1. 概要

環境変数や構成ファイルを利用することで、環境ごとの接続情報や設定値を効率的に管理できます。これにより、同じフローを複数環境で再利用しやすくなり、移行作業がスムーズになります。環境の差異を最小化し、統一的な開発運用を実現します。

6.3.2. 環境変数の利用

ソリューション内で環境変数を定義し、接続参照や固定値をまとめて管理します。たとえば、開発環境と本番環境で異なる SharePoint URL を簡単に切り替え可能です。環境変数はソリューション単位で移行できるため、再利用性が高まります。

▼ソリューションの環境変数作成画面



6.3.3. 構成ファイルの活用

JSON や Excel などの外部ファイルから設定値を読み込むことで、柔軟な構成変更が可能です。これにより、フローの修正なしで動作環境を切り替えられます。外部化された設定は、変更管理やチーム共有にも向いています。

6.4. クラウドとデスクトップのフローでデバッグ機能を利用する

6.4.1. 概要

デバッグ機能を利用することで、フローの実行中に発生する問題を効率的に特定・修正できます。処理の流れを可視化し、どのアクションが失敗したか、変数がどう変化したかを追跡できます。安定稼働に向けた品質保証のために不可欠なプロセスです。

6.4.2. クラウドフローのデバッグ

実行履歴やスコープアクションを活用して、フローの各ステップを詳細に確認します。失敗したアクションを展開し、入力・出力・応答コードなどを確認します。トリガー条件や分岐条件を修正しながら再実行テストを行うことが効果的です。

▼アクションの入力・出力

The screenshot shows the 'Array Filter Processing' (配列のフィルター処理) action in the 'Run History' (実行履歴) tab. It displays the input and output for the action.

入力 (Input):

フィールド	値
0.@odata.etag	
0.ItemInternalId	auRpdIgL Doo
0.ID	3

出力 (Output):

フィールド	値
0.@odata.etag	
0.ItemInternalId	auRpdIgL Doo
0.ID	3

6.4.3. デスクトップフローのデバッグ

デスクトップフローでは、ブレークポイントの設定や変数の監視が可能です。ステップ単位で実行を停止し、処理中の値を確認できます。これにより、複雑な UI 操作フローのトラブルを迅速に特定できます。

▼ブレークポイント

The screenshot shows a Desktop Flow (サブフロー) named 'Main'. It lists four steps:

1. 乱数の生成 (Generate Random Number) - 0 および 100 の間にある乱数を生成し、RandomNumber に保存する
2. 特別なフォルダーを取得 (Get Special Folder) - フォルダー デスクトップ のパスを取得し、SpecialFolderPath に保存する
3. フォルダー内のファイルを取得 (Get Files in Folder) - ** に一致するフォルダー SpecialFolderPath 内のファイルを取得し、Files に保存する (This step has a red dot break point and is highlighted with a yellow box)
4. テキストをファイルに書き込む (Write Text to File) - RandomNumber に SpecialFolderPath \rand.txt を書き込みます

出典：[デスクトップ フローのデバッグ - Power Automate | Microsoft Learn](#)