



Kofax RPA

Document Transformation スタート ガイド

バージョン: 11.5.0

日付: 2023-10-02

KOFAX

© 2019–2023 Kofax. All rights reserved.

Kofax is a trademark of Kofax, Inc., registered in the U.S. and/or other countries. All other trademarks are the property of their respective owners. No part of this publication may be reproduced, stored, or transmitted in any form without the prior written permission of Kofax.

目次

はじめに.....	4
関連ドキュメント.....	4
トレーニング.....	5
Kofax 製品のヘルプの入手.....	6
第 1 章：チュートリアル.....	7
事前のステップ.....	7
ロボットとタイプの作成.....	7
変換ロボットへの変数の追加.....	9
Management Console へのロボットのアップロード.....	10
Document Transformation ワークフローの作成.....	10
ドキュメントの妥当性を確認するためのステップの追加.....	15
ドキュメントからのデータ抽出.....	16
人による検証のステップの設定.....	17
ロボットが無効なドキュメントと有効なドキュメントを処理できるように設定.....	30
変換されたドキュメントの検証.....	32

はじめに

このガイドでは、Document Transformation 機能を Kofax RPA 環境で使用方法について説明するチュートリアルを提供します。このチュートリアルでは、OCR、抽出、フィールドの書式設定、検証などの機能を取り上げます。

チュートリアルでは、Kofax RPA をダウンロードしてコンピュータに正常にインストールしていることを前提としています。Kofax RPA 製品グループから Management Console および Development Database を開始した後、Windows サービス パネルで「Document Transformation - Server Scheduler Service」が開始されていることを確認します。

Document Transformation 機能の詳細な説明については、「Kofax RPA のヘルプ」を参照してください。

関連ドキュメント

Kofax RPA のドキュメント セットには次の場所からアクセスできます。¹

<https://docshield.kofax.com/Portal/Products/RPA/11.5.0-nlfihq5gwr/RPA.htm>

ドキュメント セットには、次のようなリソースがアルファベット順で含まれています。

Kofax RPA 管理者ガイド

Kofax RPA での管理タスクについて説明します。

Kofax RPA のベストプラクティス ガイド

Kofax RPA 環境でロボット ライフサイクル マネジメントを使用しながらパフォーマンスを最適化し、成功を確実にするために推奨される方法とテクニックを提供します。

Kofax RPA Desktop Automation サービス ガイド

リモート コンピューターで Desktop Automation を使用するために必要な Desktop Automation サービスを設定および管理する方法について説明します。

Kofax RPA 開発者ガイド

RoboServer でロボットを実行するために使用される Java および .NET API のプログラマー ユーザー ガイドが含まれています。また、製品で提供される Management Console REST サービスに関する情報が含まれています。

¹ オンラインのドキュメント セットにアクセスするにはインターネットに接続する必要があります。インターネットに接続せずにアクセスする方法については、『インストール ガイド』を参照してください。

Kofax RPA ロボット構築の開始ガイド

Kofax RPA を使用してロボットを構築するプロセスを実行するためのチュートリアルを提供します。

Kofax RPA Document Transformation スタート ガイド

OCR、抽出、フィールドの書式設定、検証などを含む Kofax RPA 環境の Document Transformation 機能を使用する方法について説明します。

Kofax RPA のヘルプ

Kofax RPA の使用方法について説明しています。ヘルプは、『Kofax RPA ユーザー ガイド』という PDF 形式のドキュメントとしても提供されています。

Kofax RPA インストール ガイド

Kofax RPA およびそのコンポーネントを開発環境にインストールする方法について説明します。

Kofax RPA Java API documentation (Kofax RPA Java API ドキュメント)

開発者が Kofax RPA で使用できる Kofax RPA Java API パッケージおよびクラスへのアクセスを提供します。

 Kofax RPA API は、元の製品名である「RoboSuite」に対する詳細な参照を含んでいます。RoboSuite の名前は下位互換性を確保するために残されています。API ドキュメントの中では、RoboSuite という用語は Kofax RPA と同じ意味で使われています。

Kofax RPA リリース ノート

その他の Kofax RPA ドキュメントからは入手できない最新の詳細やその他の情報が含まれています。

Kofax RPA 技術仕様

サポートされるオペレーティング システムおよびその他のシステム要件に関する情報が含まれています。

Kofax RPA アップグレード ガイド

Kofax RPA やそのコンポーネントを新しいバージョンにアップグレードする手順が含まれています。

Kofax RPA ユーザー ガイド

Kofax RPA とそのコンポーネントの使用手順が記載されています。Kofax RPA のヘルプ トピックに加えて、ヘルプに記載されていない詳細な内容が含まれています。

トレーニング

Kofax は、Kofax RPA ソリューションを最大限に活用するために、教室でのトレーニングとコンピュータでのトレーニングを提供しています。利用可能なトレーニング オプションとスケジュールの詳細については、<https://learn.kofax.com/> の Kofax 教育ポータルを参照してください。

また、<https://smarthub.kofax.com/> の Kofax Intelligent Automation SmartHub にアクセスして、追加のソリューション、ロボット、コネクタなどを見つけることもできます。

Kofax 製品のヘルプの入手

[[Kofax Knowledge Portal \(Kofax ナレッジ ポータル\)](#)] リポジトリにある記事の内容は定期的に更新され、Kofax 製品の最新情報について参照できます。製品に関してご不明の点がある場合は、Knowledge Portal (ナレッジ ポータル) で情報を検索することをお勧めします。

[Kofax Knowledge Portal] にアクセスするには、<https://knowledge.kofax.com> にアクセスしてください。

 [Kofax Knowledge Portal] は Google Chrome、Mozilla Firefox、または Microsoft Edge 向けに最適化されています。

[Kofax Knowledge Portal] は以下の内容を提供します。

- 強力な検索機能で必要な情報をすぐに見つけることができます。
[[Search \(検索\)](#)] ボックスに目的の語句を入力し、検索アイコンをクリックしてください。
- 製品情報、設定の詳細、リリース情報などのドキュメント。
記事を見つけるには、Knowledge Portal のホームページにアクセスし、製品に該当するソリューション ファミリーを選択するか、[[View All Products \(すべての製品を表示\)](#)] ボタンをクリックします。

Knowledge Portal のホームページからは、次の操作を実行できます。

- Kofax Community (Kofax コミュニティ) へのアクセス (全カスタマー)。
[[Resources \(リソース\)](#)] メニューで、[[Community \(コミュニティ\)](#)] リンクをクリックします。
- Kofax Customer Portal (Kofax カスタマー ポータル) へのアクセス (一部のカスタマーのみ)。
[[Support Portal Information \(サポート ポータルの情報\)](#)] ページに移動し、[[Log in to the Customer Portal \(カスタマー ポータルにログイン\)](#)] をクリックします。
- Kofax Partner Portal (Kofax パートナー ポータル) へのアクセス (一部のパートナーのみ)。
[[Support Portal Information](#)] ページに移動し、[[Log in to the Partner Portal \(パートナー ポータルにログイン\)](#)] をクリックします。
- Kofax サポート コミットメント、ライフサイクル ポリシー、電子フルフィルメントの詳細、セルフサービス ツールへのアクセス。
[[Support Details \(サポートの詳細\)](#)] ページに移動し、適切な記事を選択します。

第 1 章

チュートリアル

このガイドは、製品に付属しているサンプル ロボットのような Document Transformation ロボットを作成する方法について説明するステップバイステップのチュートリアルです。 *DocumentTransformationInvoiceExampleBasic* という名前のサンプル ロボットには、変換可能な .tif ファイルがすでに含まれています。

製品に付属している *DocumentTransformationInvoiceExampleBasic* ロボットは、ロボットとタイプを新しく作成せずにテストすることができます。 **Design Studio** の **Examples** フォルダでロボットを開いて、タイプ *invoice* と *DocumentTransformationInput* を開き、このガイドに従ってロボットの構築方法を確認できます。

ロボットを最初から作成する場合は、このガイドの手順を使用して、必ず サンプルとは異なる一意の名前をロボットとタイプに割り当ててるようにしてください。

このチュートリアルでは最も一般的に使用される Document Transformation 関数のいくつかを、主要 4 パート構成で取り上げます。

1. 変換ロボット (ベーシック エンジン ロボット  を使用)、変換ワークフロー (ロボット  を使用)、およびデータを格納するタイプを作成します。
2. 「ロボットを呼び出す」ステップを使用して、ベーシック エンジン ロボット  からロボット  を呼び出します。
3. ロボット  で Document Transformation ワークフローを作成します。これには、[Document Transformation] ステップの設定、.tif ドキュメントからのデータ抽出、および変換されたドキュメントの人による検証と人によらない検証の設定が含まれます。
4. 変換されたドキュメントを検証します。

事前のステップ

ロボットとタイプの作成

最初に、ベーシック エンジン ロボットを作成し、変換ワークフローを構築するために必要なロボットを作成する必要があります。また、抽出されたデータを保存するためにタイプを作成する必要があります。青いアイコン  で識別されるベーシック エンジン ロボットとは対照的に、ロボットは緑色のアイコン  で識別されます。

1. ベーシック エンジン ロボットの作成 
 - a. **Design Studio** で [ファイル] > [新しいベーシック エンジン ロボット] をクリックします。

- b. ロボットに **DocumentTransformationInvoiceExampleBasic** という名前を付け、プロジェクトを選択し、[次へ]>[次へ]の順にクリックします。
デフォルトで [スマート再実行 (フル)] 実行モードが選択されます。
- c. [終了] をクリックします。
デフォルトでは、新しく作成されたロボットでは終了ステップが選択されます。
- d. 新規作成ロボットに [アクション ステップ] を挿入します。
- e. ステップの名前を「**Transform Document**」に変更します。
- f. 変更を保存します。

2. ロボットの作成

- a. [ファイル]>[新規ロボット] をクリックします。
- b. ロボットに **DocumentTransformationInvoiceExample** という名前を付け、[終了] をクリックします。
エディター ウィンドウの新しいタブに新しいロボットが表示されます。

3. ベーシック エンジン ロボットからのロボットの呼び出し

- a. **DocumentTransformationInvoiceExampleBasic** ベーシック エンジン ロボットが表示されているタブを開きます。
- b. 挿入したステップで、[アクション] タブの [アクションを選択] をクリックし、[ロボットを呼び出す] を選択します。
- c. [ロボット] ドロップダウン リストで [**DocumentTransformationInvoiceExample**] というロボットを選択します。
- d. 変更を保存します。

4. 新しいタイプを作成

このロボットを使用して、ベンダーの ID、請求書番号、請求書日付、合計を抽出します。抽出したデータを格納するには、以下のタイプを作成します。

- a. [ファイル]>[新しいタイプ] をクリックします。
- b. タイプに **invoice** という名前を付け、プロジェクトを選択し、[終了] をクリックします。
- c. プラス記号をクリックして新しい属性をタイプに追加します。次の属性を追加して、そのタイプを指定します。

名前	属性の種類
VendorID	Short Text
InvoiceNumber	Short Text
InvoiceDate	Short Text
Total	Number

- d. **DocumentTransformationInput** という名前のタイプを別に作成します。
- e. 次の属性を追加して、そのタイプを指定します。

名前	属性の種類
doc	Binary

f. 変更を保存します。

終了したら次のセクションに進みます。

変換ロボットへの変数の追加

この手順では、コンプレックス タイプの変数とシンプル タイプの変数を追加します。複雑な変数には複数の名前が付いた値を含めることができます。これは請求書のデータのようなさまざまなデータの抽出に有用です。シンプル変数には、テキスト文字列のような単一の値のみを含めることができます。

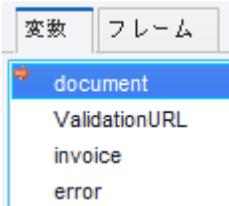
1. **Design Studio** の [変数] フィールドで右クリックし、[コンプレックス タイプの変数を追加] > **DocumentTransformationInput** の順にクリックします。

a. 新しいダイアログ ボックスで「**DocumentTransformationInput**」変数の名前を「**document**」に変更し、[パラメータとして使用] を選択します。

b. [ロード] をクリックし、変換する領収書ドキュメントを開きます。サンプルプロジェクトの Kofax RPA [Examples] フォルダに移動し、ファイル **InvoiceExampleDocument.tif** を選択します。この領収書は、組み込みの Invoice VAT Transformation プロジェクトで読み取ります。

2. [変数] フィールドで右クリックし、[コンプレックス タイプの変数を追加] > **invoice** の順にクリックします。

3. [変数] フィールドで右クリックし、[シンプル タイプの変数を追加] > **Short Text (simple)** の順にクリックします。新しい変数に「**ValidationURL**」という名前を付けます。
4. [変数] フィールドで右クリックし、[シンプル タイプの変数を追加] > **Long Text (simple)** の順にクリックします。新しい変数に **[Error]** という名前を付けます。
5. 変更を保存します。



6. **Transform Document** ステップの [アクション] タブを開き、次のように変数を構成します。
 - a. [入力値] セクションで、プラス記号をクリックします。
ドロップダウン リストで、[変数] をクリックして **[document]** を選択します。
 - b. **[OK]** をクリックします。
 - c. [出カマッピング] セクションで、「**invoice**」、「**ValidationURL**」、「**error**」変数を追加します。

終了したら次のセクションに進みます。

Management Console へのロボットのアップロード

[Projects] ツリービューで、**DocumentTransformationInvoiceExampleBasic** ロボットを右クリックし、[アップロード] をクリックしてロボットを Management Console にアップロードします。

終了したら次のセクションに進みます。

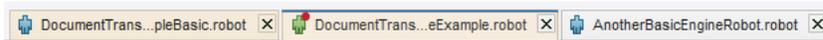
Document Transformation ワークフローの作成

この手順では、変数を設定して、Document Transformation ステップを設定し、変換されたドキュメントを待機するためのガードを追加します。

ベーシック エンジン ロボットの編集と実行を開始するには、アプリケーション ビューまたはツールバーの [実行の準備] をクリックして、ロボットの実行準備を行う必要があります。このアクションをクリックすると、ロボットが実行モードになり、編集中にロボットを実行できるようになります。アクション ステップは、ロボット ワークフローに挿入した直後に実行することができ、すぐに結果が表示されます。ベーシック エンジン ロボットの実行準備ができていない場合でもステップの追加などの基本的な一部の基本的な編集を実行できますが、ステップを実行して結果を表示することはできません。

⚠ 実行権限を保持できるベーシック エンジン ロボットは一度に 1 つのみであるため、ロボット間で実行権限を引き継ぐには、必要なロボットが配置されたタブを開いて、[実行の準備] をクリックします。

ベーシック エンジン ロボットが実行権限を保持している場合は、このロボットのエディター タブがハイライト表示されます。ベーシック エンジン ロボットがロボットを呼び出している場合は、次のように、両方のロボットのタブがハイライト表示されるので便利です。現在実行されているロボットには赤い点が付いています。



1. 実行が許可されている場合は、ロボット  を開きます。この操作を行うには、「ロボットを呼び出す」ステップのワークフローを実行してから、ツールバーの [ロボットにステップ]  をクリックします。

DocumentTransformationInvoiceExample ロボットのタブが開き、エディターがアクティブになります。

2. ワークフローで、次のステップに従います。
 - a. [入力値] ボックスが次の方法で設定されていることを確認します。



- b. この変数には、検証後の変換されたドキュメントが含まれます。
- c. [変数] ボックスで、次の変数を追加し、各タイプを指定します。

(x) 変数

Invoice

タイプ
invoice

名前
Invoice

ValidationURL

タイプ
Text

名前
ValidationURL

Valid

タイプ
Boolean

名前
Valid

Error

タイプ
Text

名前
Error

d. ワークフローで、**Return** ステップを展開し、以下の変数を追加します。

1. **Invoice** -必ず大文字で始めるようにしてください。
2. **ValidationURL**
3. **Error**

リターン

値

=Invoice

=ValidationURL

=Error

3. [Document Transformation ステップ] 追加するには、リターン ステップの前のフローポイント (小さな円) を右クリックして、アプリケーション > **Document Transformation** をクリックします。このステップでは、データを `InvoiceExampleDocument.tif` から抽出します。

i このセクションの後半にある Document Transformation ステップを実行すると、文書の変換が失敗したというエラーと変換プロジェクトを参照するメッセージが表示される場合があります。これは、Document Transformation トレーニング ファイルを保存するフォルダがないことを意味する場合があります。

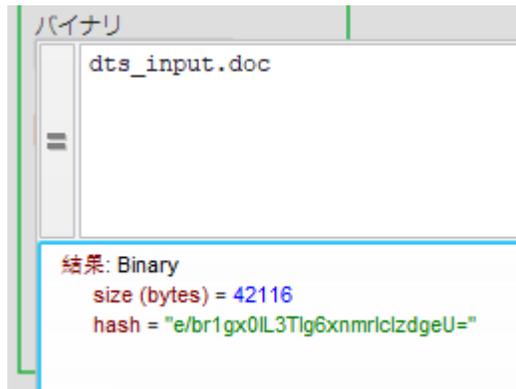
この問題を解決するには、「**Invoice VAT**」プロジェクトの [Project Builder] 内で [プロジェクト設定] を開き、[一般] タブで [OK] をクリックします。フォルダが存在しない場合、作成することを求める通知を受け取ります。続行するには、[はい] をクリックします。

詳細については、『Kofax RPA インストール ガイド』の「オンライン学習の設定」セクションを参照してください。

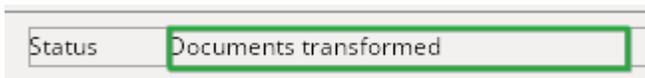
4. プロパティを以下のように設定します。これらのプロパティの詳細については、『Kofax RPA のヘルプ』の「Document Transformation」を参照してください。



- a. [サービス URL] で、Document Transformation Service を実行しているコンピュータの URL を指定します。この例のロボットでは、URL は `http://127.0.0.1:50081` となります。
- b. [ドキュメント ソース] で、[バイナリ変数] 選択し、[バイナリ] フィールドをクリックして、`dts_input.doc` を入力します。変数を評価できるように、左側に等号が表示されていることを確認します。



- c. [検証 URL] を選択して、シンクライアント サービスの URL を指定します。このプロパティは、処理されたドキュメントを検証の目的で送信するために必要です。この例のロボットでは、URL は次のようになります。http://127.0.0.1:50082
 - d. [コールバック] を選択し、変換ロボットに関する情報を指定します。シンクライアント サービスは、ドキュメントの検証後にこのロボットを呼び出します。検証が完了すると、ロボットは実行のために、Management Console のキューに登録されます。この例のロボットの **Document Transformation** ステップでは、[ロボットプロジェクト] と [ロボット名とパス] が次のように設定されます。
 - デフォルトのプロジェクト
 - Examples/Robots/DocumentTransformationInvoiceExampleBasic.robot
 - e. [ステップ オーバー] をクリックしてこのステップを実行します。ステータスが「Waiting for transformed documents.」に変更されます。「Documents transformed.」に変更されるまで待機します。
5. 「Documents transformed」をクリックして、変換されたドキュメントを待機するためのガードを追加します。

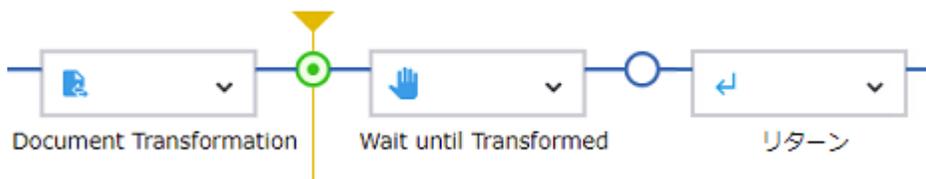


- a. ガードが正しいテキストを検索できるようにするため、XML ツリーの以下のラベルを展開します。

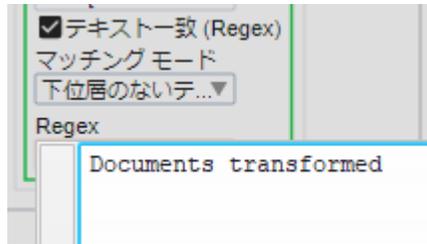
```
<label name="status_message" der_x="494" der_y="56" der_width="230" der_height="20">Documents transformed</label>
```

テキスト「Documents transformed」を右クリックし、[ガード] > [該当するロケーション] の順にクリックします。

ガードステップがワークフローに挿入されます。ステップ名を「**Wait until Transformed**」に変更します。



- b. **Wait until Transformed** ステップで、[該当するロケーション] ガードを展開し、[テキスト一致 (Regex)] を選択して「Documents transformed」と入力します。



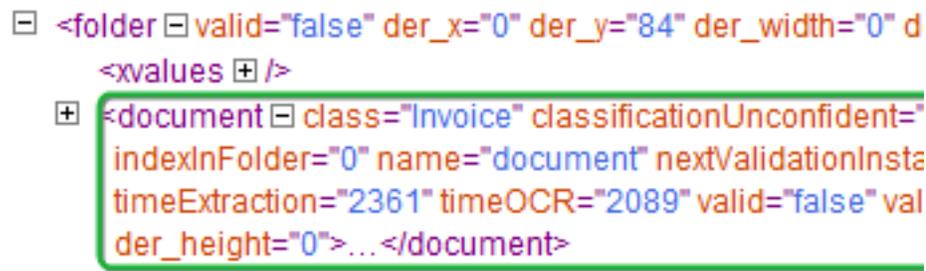
- c. [ステップ オーバー] をクリックしてこのステップを実行します。
d. 変更を保存します。

終了したら次のセクションに進みます。

ドキュメントの妥当性を確認するためのステップの追加

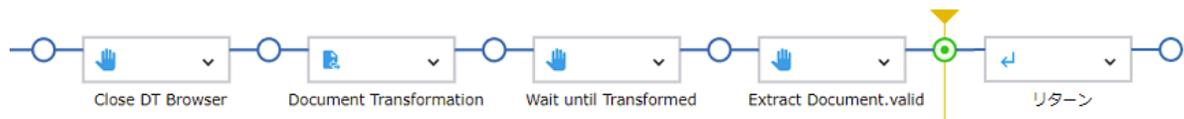
変換したドキュメントが妥当かどうかを確認するには、次のステップを Document Transformation ワークフローに追加します。ドキュメントが妥当でないようであれば、人による検証のために送信できます (人による検証のステップの設定を参照)。

- XML ツリービューで **document** 要素を見つけて右クリックし、[ここから値を抽出] > [属性] > **valid Into** > **Valid: Boolean** の順にクリックします。



新しいステップがワークフローに挿入されます。ステップ名を「**Extract Document.valid**」に変更します。

- [ステップ オーバー] をクリックしてこのステップを実行します。



- 変更を保存します。

終了したら次のセクションに進みます。

ドキュメントからのデータ抽出

この手順では、データを InvoiceExampleDocument.tif から変数 VendorID、InvoiceNumber、InvoiceDate、Total に抽出します。最後に、[値返却] ステップをロボットに追加すると、抽出された値が返されます。

1. XML ツリービューで、**application/folder/document** の下にある **fields** 要素を展開します。

```
<dts ⊕ >
  ⊖ <application ⊕ >
    ⊕ <toolbar ⊕ >...</toolbar>
    ⊖ <folder ⊕ >
      <xvalues ⊕ />
      ⊖ <document ⊕ >
        ⊕ <fields ⊖ der_x="0" der_y="84" der_width="0" der_height="0">...</fields>
        ⊕ <xvalues ⊕ >...</xvalues>
        ⊕ <pages ⊕ >...</pages>
```

2. 値を抽出するには

- a. **VendorID** の値と alternative を確認するには、対応する 2 番目のフィールドを展開します。
- b. **field** 要素を右クリックし、[ここから値を抽出] > [属性] > **validatedText Into** > **Invoice: invoice** > **VendorID: Text** の順にクリックします。

```
⊖ <field ⊖ name="VendorID" page="0" confidence="0.613246" errorD
  formattingFailed="false" valid="true" validatedText="198000" verifie
  198000
  ⊕ <alternative ⊕ >198000</alternative>
```

i テキストの 198000 をクリックしないようにしてください。

新しい [値を抽出] ステップがワークフローに挿入されます。ステップ名を「**Extract Vendor ID**」に変更します。

- c. このアクションをフィールド 17 の **InvoiceNumber** とフィールド 18 の **InvoiceDate** に対して繰り返します。

新しい [値を抽出] ステップがワークフローに挿入されます。各ステップ名を「**Extract Invoice Number**」および「**Extract Invoice Date**」に変更します。

- d. 最後から 3 番目のフィールド **TotalAmount** を右クリックし、[ここから値を抽出] > [属性] > **doubleValue Into** > **Invoice: invoice** > **Total: Number** をクリックします。

変換プロジェクトはこのフィールドに数値フォーマッタを持つため、フィールドは数値に変換されます。

[値を抽出] ステップがもう 1 つワークフローに挿入されます。ステップ名を「**Extract Total Number**」に変更します。

3. [実行を開始] ▶ をクリックし、ワークフローを最後まで実行します。

[値を抽出] ステップが実行されると、抽出された値を確認することができます。右側の [状態] ペインで、[変数] 分岐を展開します。

4. 変更を保存し、ロボットからステップ アウトします。

ロボットからステップ アウトして、ベーシック エンジン ロボットでの作業に切り替えるには、ワークフロー全体を実行し、ツールバーの [ステップ アウト]  をクリックします。ベーシック エンジン ロボットで「ドキュメントを変換」ステップが実行済みと表示されるようになりました。

これで、Document Transformation ロボットは領収書の詳細のすべてに対応できるようになりました。Design Studio の [変数] セクションで **invoice** 変数を確認します。

5. Design Studio で、別の [アクション ステップ] をベーシック エンジン ロボットに追加し、これを [値返却] ステップにします。ツールバーで、[選択したステップの後にステップを挿入]  > [アクション ステップ] の順にクリックし、[アクション] タブで [値返却] を選択します。

[基本] タブの [変数] リストで [**invoice**] を選択します

この方法で、ロボット デバッガに invoice パラメータを表示することができます。

6. 変更を保存します。

これで、Management Console で実行される有効なロボットが作成できました。このロボットは、Document Transformation クライアントによって呼び出されます。

 変換が完了する前に、ドキュメントのバックアップ コピーがあることを確認してください。変換が終了した場合、またはエラーが発生した場合には、ドキュメントおよびデータは格納されません。

終了したら次のセクションに進みます。

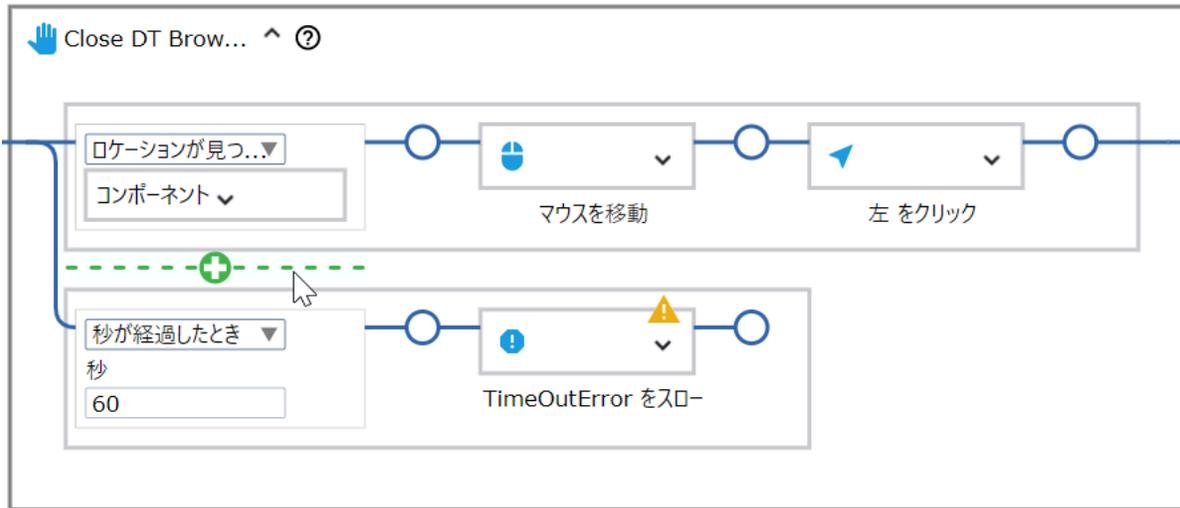
人による検証のステップの設定

この手順では、人による検証のためにドキュメントを送信するシナリオを処理するガードを設定し、人による検証のワークフローに特別な条件分岐を追加します。

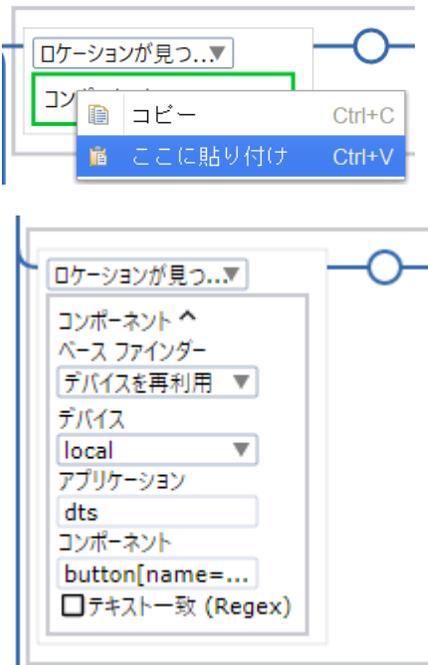
Transform Document ステップを再度開く必要があります。ステップを再度開く前にロボットを更新し、前回のロボットの実行で開いた Document Transformation ブラウザ ウィンドウが、現在は閉じていることを確認します。

ヒント: または、クリック ステップを追加してロボットの実行終了後にブラウザ ウィンドウを閉じるようにすることもできます。これを実行するには、**DocumentTransformationInvoiceExample** ロボットでブラウザの [閉じる] ボタンを右クリックし、[クリック] > [左] の順にクリックします。新しいステップがワークフローに追加されます。

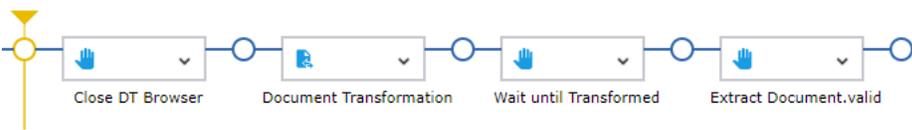
新しいステップを展開し、新しいガードを追加するには中間にある緑のプラス アイコンをクリックします。



新しいガードを [該当しないロケーション] に設定し、[該当するロケーション] ガードから [コンポーネント] を [該当しないロケーション] ガードの [コンポーネント] ボックスにコピーします。



作成したステップは、Document Transformation ブラウザが開いていた場合にはすぐに閉じ、それ以外の場合には実行を続行します。ステップの名前を「Close DT Browser」に変更します。



完了後に、以下の手順に進んで人による検証を設定します。この手順では、検証されたドキュメントが Document Transformation クライアントから検証コールバックを経由して戻されたケースを扱います。

1. **Document Transformation** ステップの後にあるフロー ポイントをダブルクリックします。

Wait until Transformed ステップの「該当するロケーション」ガードが、変換されたドキュメントを待機します。

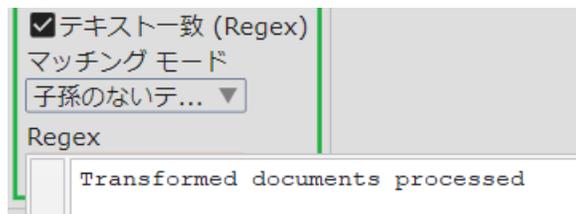
Document Transformation クライアントは、人による検証の後に、[コールバック] オプションで指定されたロボットを呼び出すと、変換されたドキュメントを .zip ファイルとしてロボットの **document.doc** 変数に渡します。 .zip ファイルには、「project」フォルダと、config.xml、document.tif、document.xdc、Folder.xfd、tree.dat などのいくつかのファイルが含まれているはずで

Document Transformation ブラウザで .zip ファイルを開くと、ステータスが「Transformed document processed」に変更されます。

2. 変換されたドキュメントが処理されるまで待機するようにガードを設定するには、**Wait until Transformed** ステップで、追加の [該当するロケーション] ガードを作成し、最初の [該当するロケーション] ガードの [コンポーネント] ボックスをコピーして 2 番目のガードの [コンポーネント] ボックスに貼り付けます。



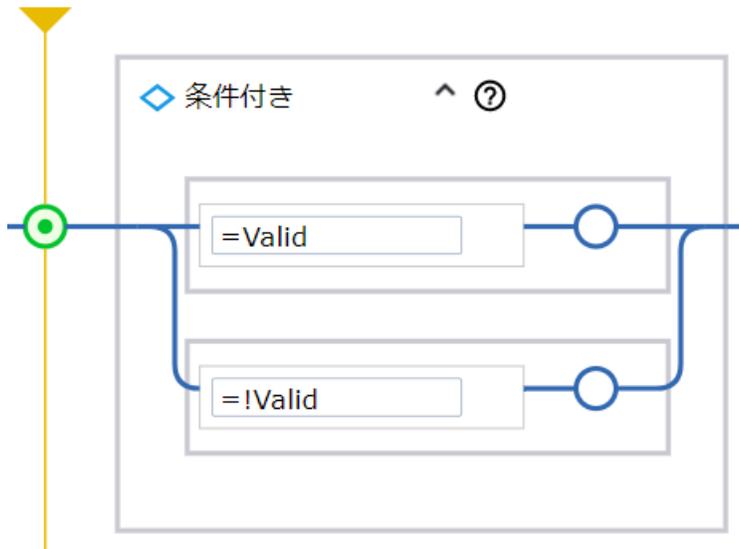
2 番目のガードの [テキスト一致 (Regex)] を「**Transformed documents processed**」に変更します。



2 つのガードが設定され、ワークフローでは人による検証と、人を必要としない検証の 2 つのシナリオを処理できるようになりました。

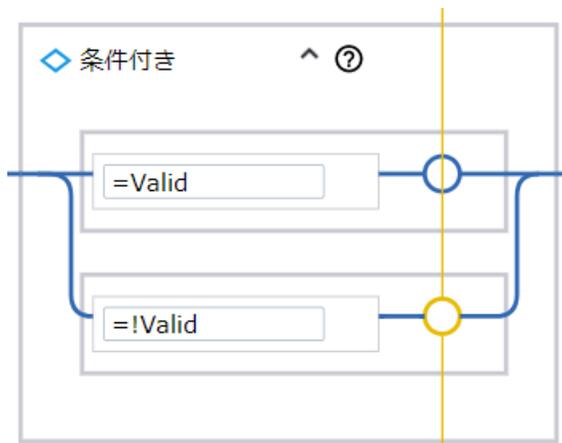
i このステップでは、オプションで追加のガードを設定して、変換中に発生する可能性のあるエラーを処理できます。詳細については、[エラー処理の設定](#)を参照してください。

3. このポイントに対して実行する **Extract Document.valid** ステップの後のフローポイントをダブルクリックします。
4. 人による検証のために特別な分岐を追加します。フローポイントを右クリックし、[条件と制御] [条件付き] をクリックします。新しいステップがワークフローに挿入されます。
 - a. 次の2つの条件を **Conditional** ステップに追加します。

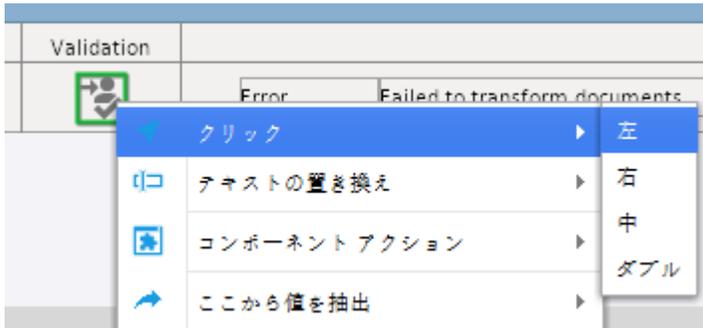


=!Valid の感嘆符は「not」を表します (「有効でない」)。

- b. [ステップ] をクリックし、**Conditional** ステップに入ります。
この時点でロボットのドキュメントは有効ではないため、2番目の条件 (分岐) が実行されます。



5. 次に、ドキュメントを Document Transformation クライアントに送信する必要があります。[レコーダービュー] で、検証ボタンを右クリックし、[クリック] > [左] の順にクリックします。

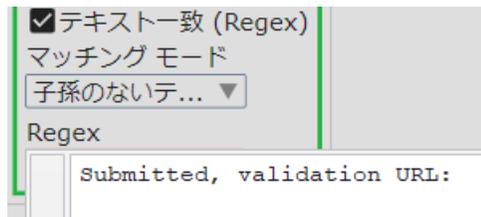


Conditional ステップの 2 番目の分岐に新しいステップが挿入されます。ステップ名を「**Send to Validation**」に変更します。[ステップ オーバー] をクリックしてステップを実行します。

- a. ステータスが「**Submitted, validation URL:**」に変更されたらクリックし、[ガード] > [該当するロケーション] の順にクリックします。

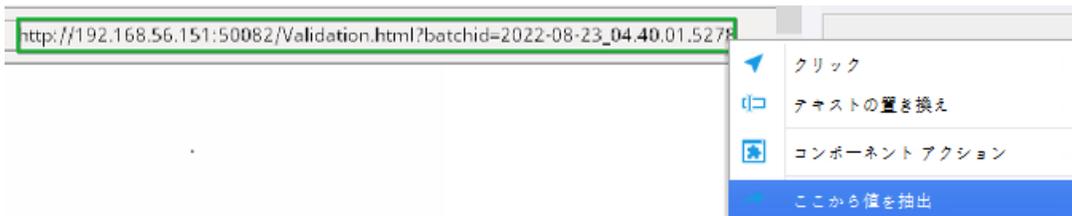
Conditional ステップの 2 番目の分岐に新しいステップが挿入されます。ステップ名を「**Wait for URL**」に変更します。

- b. **Wait for URL** ステップで、[コンポーネント] ボックスを展開し、[テキスト一致 (Regex)] を選択し、「**Submitted, validation URL:**」と入力します。

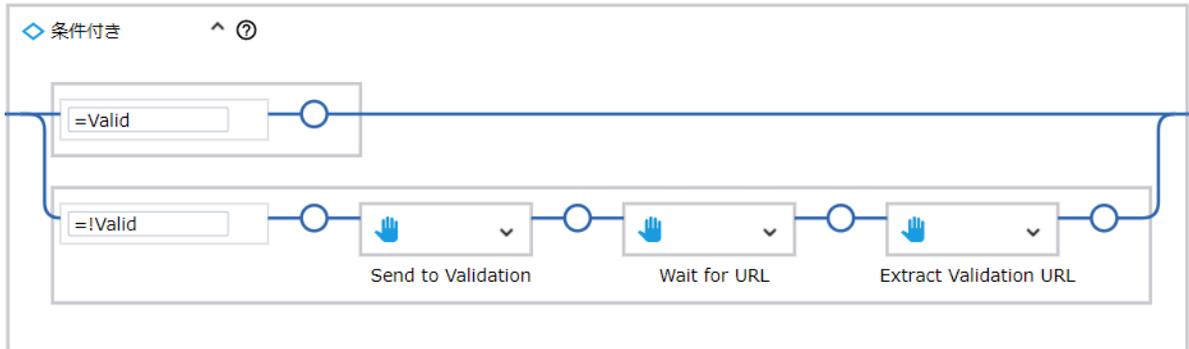


- c. [ステップ オーバー] をクリックします。

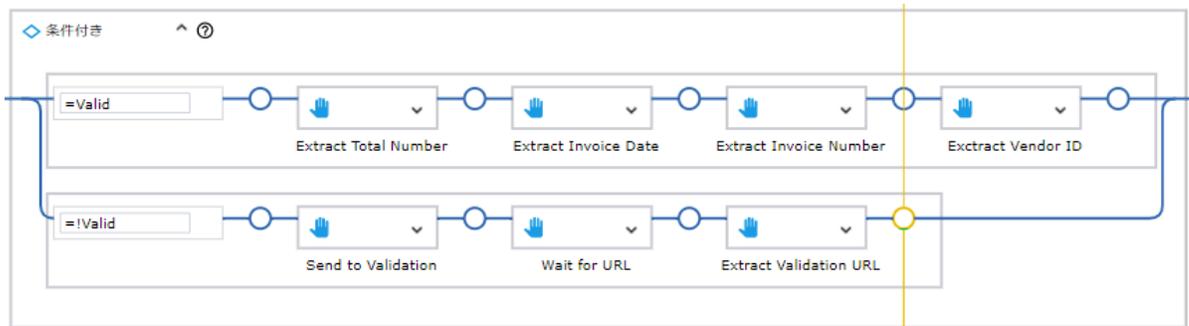
6. 以下の URL をクリックし、[ここから値を抽出] > [テキスト] > **ValidationURL: Text** の順にクリックします。



「値を抽出」ステップが **Conditional** ステップ内に挿入されます。ステップ名を「**Extract Validation URL**」に変更し、実行します。



7. **Conditional** ステップの後の 4 つの [値を抽出] ステップを選択し、**Conditional** ステップの =Valid 分岐にカット アンド ペーストします。



8. 変更を保存します。

これで変換ワークフローは終了です。これでワークフローには 3 つの可能なパスがあることとなります。

- 新しいドキュメントは抽出と分類の処理のために Document Transformation サービスに送信され、「有効」が戻る。ドキュメントには人による検証は不要である。
- 新しいドキュメントは抽出と分類の処理のために Document Transformation サービスに送信され、「無効」が戻る。ドキュメントには人による検証が必要である。
- 人による検証後にドキュメントの .zip パッケージが Document Transformation クライアントから送信され、「有効」が戻る。検証 URL が取得される。

[実行を開始] ▶ をクリックし、ワークフローを最後まで実行します。完了後に、ロボットをステップアウトしてロボットが無効なドキュメントと有効なドキュメントを処理できるように設定に進みます。

- ロボットからステップアウトして、ベーシックエンジンロボットに切り替えるには、ワークフロー全体が実行された後に、ツールバーの [ステップアウト] をクリックします。ベーシックエンジンロボットで「ドキュメントを変換」ステップが実行済みと表示されるようになりました。
- 最後まで実行しないで、または結果を返さないでロボットを終了するには、[実行を開始] ▶ をクリックする代わりに、ツールバーで [ロボットの終了] をクリックします。ベーシックエンジンロボットが配置されたタブが開きます。「ドキュメントを変換」ステップが未実行と表示されます。

エラー処理の設定

オプションでエラー処理を設定して、変換プロセス中に問題が発生した場合にエラーが報告され、エラーに関する情報が変数に保存されるようにできます。後で、エラー情報を含む電子メールを人に送信するように設定できます。

次の手順は、**Wait until Transformed** ステップでエラー処理のガードを設定する方法を説明します。インストールで提供される *DocumentTransformationInvoiceExampleBasic* ロボットには、**Close DT Browser** および **Extract Document.valid** ステップに「時間経過」ガードが追加されています。この2つのステップは同様の方法で設定されています。

1. **Wait until Transformed** ステップで、2 番目の [該当するロケーション] ガードの下に別の [該当するロケーション] ガードを作成し、2 番目のガードから [コンポーネント] ボックスをコピーして、3 番目のガードの [コンポーネント] ボックスに貼り付けます。
2. コピーした [コンポーネント] ボックスを展開し、次の変更を行います。
 - [コンポーネント] フィールドで、値を `label[name="status_title"]` に変更します。
 - [テキスト一致 (Regex)] を **Error** に変更します。

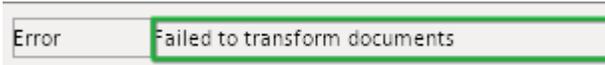
このガードにより、ワークフローは別のシナリオを処理します (エラーが報告された場合)。他の2つのシナリオについては、[人による検証のステップの設定](#)で説明されています。

3. [該当するロケーション] ガードの横にあるフローポイントを右クリックし、条件と制御 > グループ化をクリックします。グループの名前を「**Extract Error**」(エラーの抽出)に変更します。ステップを展開し、[変数] ボックスを展開して、次のようにテキスト タイプの2つの変数を追加します。



これらの変数には、エラーが発生した場合にエラー メッセージ名と情報が格納されます。

4. このポイントに対して実行する [変数] ボックスの横にあるフロー ポイントをダブルクリックします。
5. [レコーダー ビュー] で、[Failed to transform documents] ボタンを右クリックし、[ガード]>[該当するロケーション] をクリックします。



新しいガードがワークフローに挿入されます。これは、エラー メッセージ名を抽出するガードです。名前を「**Extract Message**」に変更します。

- [該当するロケーション] ボックスの横のフロー ポイントを右クリックし、[割り当てと変換]>[値を抽出] の順にクリックして、[値を抽出] ステップを次のように設定します。



このステップでは、エラーが報告された場合、エラー名が変数 **ErrorMessage** に保存されます。

- [該当するロケーション] ボックスの下のプラス記号をクリックして、新しいガードを追加します。



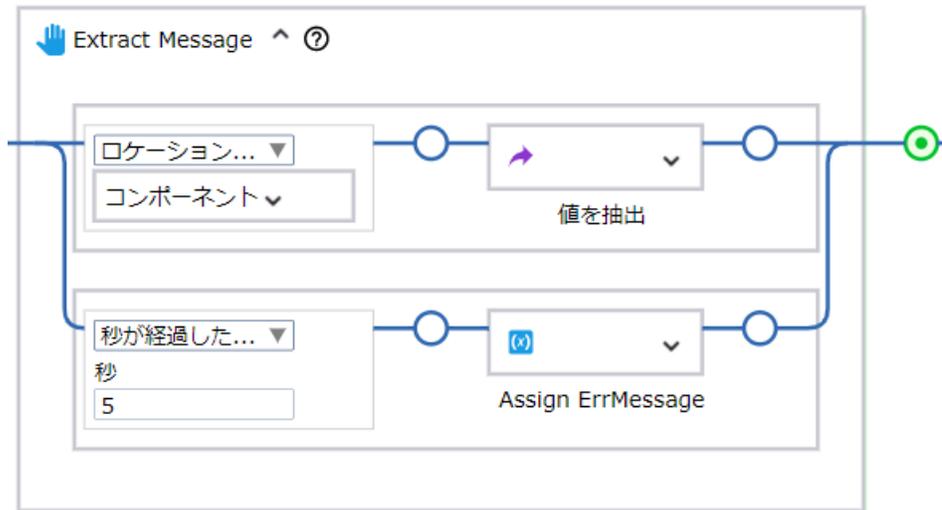
デフォルトでは、「時間経過」タイプのガードが挿入されます。5 秒に設定します。

- [時間経過] ガードの横にあるフローポイントを右クリックし、[条件と制御] から [割り当て] ステップを追加し、名前を「**Assign ErrorMessage**」に変更します。
割り当てステップを次のように設定します。



この2つのステップで不明なエラーが報告された場合、該当するエクスペッションが変数 **ErrorMessage** に保存されます。

6. このポイントに対して実行する **Extract Message** ステップの横にあるフローポイントをダブルクリックします。



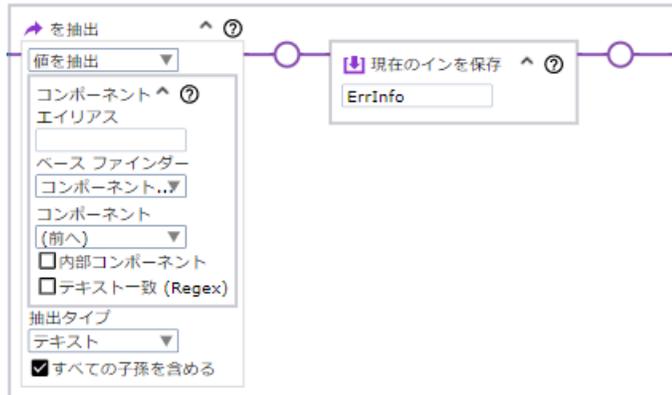
7. [レコーダービュー] で、エラーの説明が示されているボタンを右クリックし、[ガード]> [該当するロケーション] をクリックします。

Error: connect ECONNREFUSED 127.0.0.1:50081

Document Transformation Service がターゲット コンピュータで実行されていない場合に表示される接続拒否エラーです。

新しいガードがワークフローに挿入されます。エラーに関する情報を抽出するガードです。名前を「**Extract Info**」に変更します。

- [該当するロケーション] ボックスの横のフローポイントを右クリックし、[割り当てと変換] から [値の抽出] ステップを追加して、次のように設定します。



このステップでは、エラー メッセージ情報が変数 **ErrInfo** に保存されます。

- [該当するロケーション] ボックスの下のプラス記号をクリックして、新しいガードを追加します。
「時間経過」を 5 秒に設定します。
- [時間経過] ガードの横にあるフロー ポイントを右クリックし、[割り当てと変換] から [割り当て] ステップを追加し、名前を「**Assign ErrInfo**」に変更します。
割り当てステップを次のように設定します。



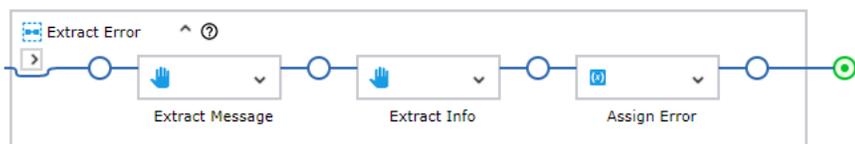
この 2 つのステップで不明なエラーが報告された場合、該当するエクスペッションが変数 **ErrInfo** に保存されます。

- このポイントに対して実行する **Extract Info** ステップの横にあるフロー ポイントをダブルクリックします。
- [**Extract Info**] ステップの横にあるフロー ポイントを右クリックし、[割り当てと変換] から [割り当て] ステップを追加し、名前を「**Assign Error**」に変更します。
 - [変数] フィールドに「**Error**」と入力します。[Document Transformation ワークフローの作成](#)で指定した変数の 1 つです。
 - [エクスペッション] フィールドに、次のエクスペッションを入力します。**ErrorMessage + ":" + ErrInfo**
左の灰色のバーをクリックすると等号が表示され、エクスペッションを評価できます。



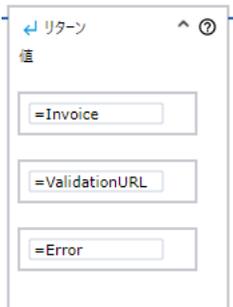
このステップでは、「**Error**」変数はエラー メッセージ名とエラー情報を返します。

Extract Error グループは、次のようになります。

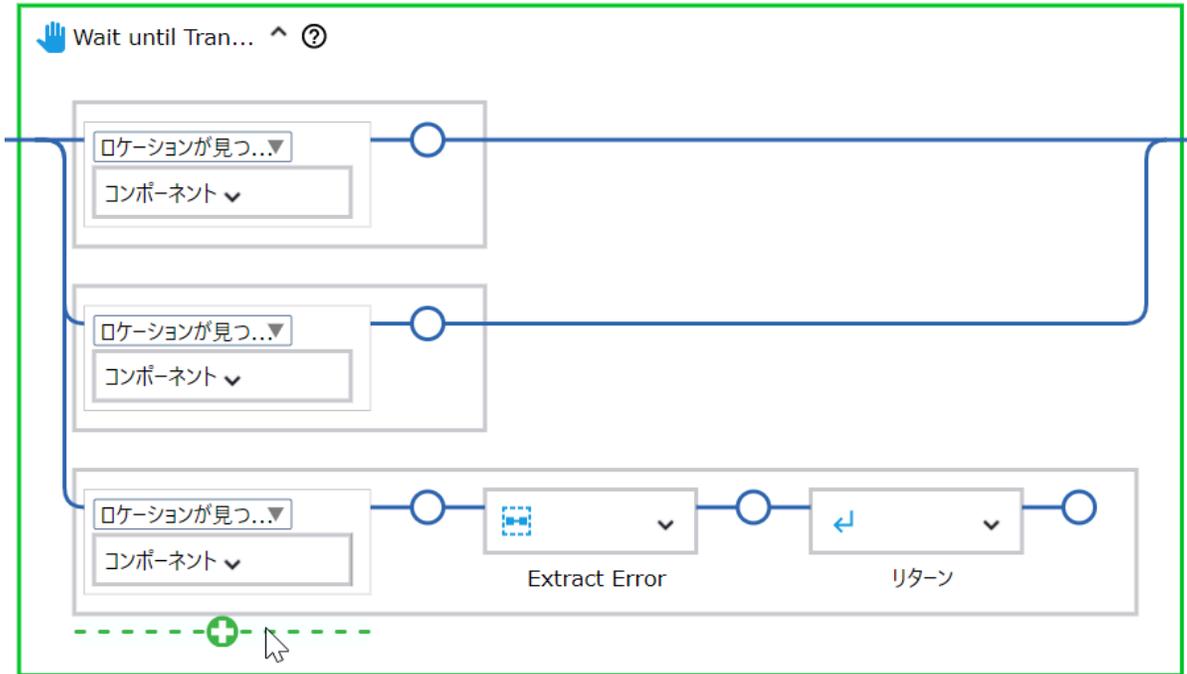


10. グループの横にあるフロー ポイントを右クリックし、[条件と制御] から [リターン] ステップを追加します。

次のように設定します。



11. 次に、3 番目の [該当するロケーション] ガードの下にあるプラス記号をクリックして、別のガードを追加します。



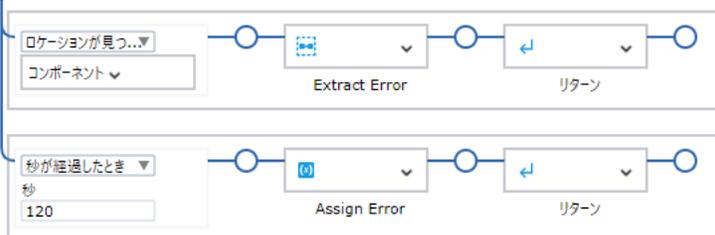
- 「時間経過」を 120 秒に設定します。
- このガードの横に、[割り当てと変換] から [割り当て] ステップを追加し、名前を「Assign Error」に変更します。
- [変数] フィールドに「Error」と入力します。
- [エクスペッション] フィールドに、次のテキストを入力します。

Waiting for transformation timed out

- **Assign Error** ステップの後、ステップ 10 と同じ [リターン] ステップを追加します。ステップをコピーして貼り付けることができます。

これらのステップを使用して、変換プロセスに時間がかかる場合、上記で指定したエクスペッションが戻り変数「Error」に格納されます。

Wait until Transformed ステップのエラー処理部分は、次のようになります。



ロボットが無効なドキュメントと有効なドキュメントを処理できるように設定

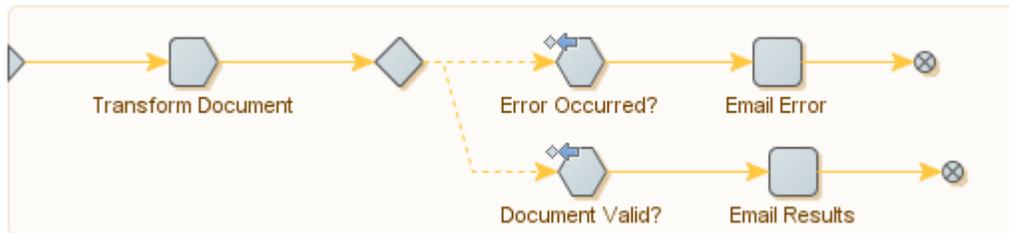
この手順では、Design Studio のロボットに分岐を設定し、3つのシナリオ (ドキュメントが有効な場合と無効な場合、または変換中にエラーが発生した場合) を処理できるようにします。ドキュメントが有効な場合、ロボットは Transform Document ステップで抽出した値を電子メールで人に送信します。変換されたドキュメントが無効な場合、人による検証のために送信されます。エラーが発生した場合、エラー情報を含む電子メールが人に送信されます。

1. **Design Studio** で Transform Document ステップの後に [トライ ステップ] を追加します。
「ドキュメントを変換」ステップをクリックし、ツールバーで [選択したステップの後にステップを挿入]  > [トライ ステップ] の順にクリックします。
2. 最初の分岐に [アクション ステップ] を追加し、[値判定] ステップにします。
[値返却] ステップを右クリックし、[ステップを前に挿入] > [アクション ステップ] の順にクリックし、[アクション] タブ上で [値判定] を選択します。
 - a. [条件] セクションで「**error==""**」と入力します。直線型の引用符を使用します。
 - b. [If] および [Do] を [条件が満たされています] および [[エラー処理] 時に指定されたとおり] にそれぞれ設定します。
 - c. [エラー処理] タブの [Then] リストで [次のトライステップへ移動] を選択します。
 - d. ロボットで、[値判定] ステップの名前を「**Error Occurred?**」に変更します。
3. [Error Occurred?] ステップの後に [アクション ステップ] を追加し、[メール送信] ステップにします。
[アクション] タブで、[出力] > [メール送信] を選択します。
4. [メール送信] ステップを次のように設定します。
 - a. [送信元アドレス] フィールドと [宛先アドレス] フィールドに、送信者と受信者の電子メール アドレスをそれぞれ入力します。
 - b. [メッセージ] フィールドに [エクスプレッション] を設定し、以下のエクスプレッションを追加します。
"Something went wrong during transformation or subsequent processing.\n" + error
 - c. [メッセージ タイプ] を [テキスト] に設定します。
 - d. [件名] フィールドに、次のようなテキストを入力します。"Transformation has failed."
 - e. [サーバー] タブで、SMTP サーバー設定を行います。
 - f. ロボットで、ステップの名前を「**Email Error**」に変更します。



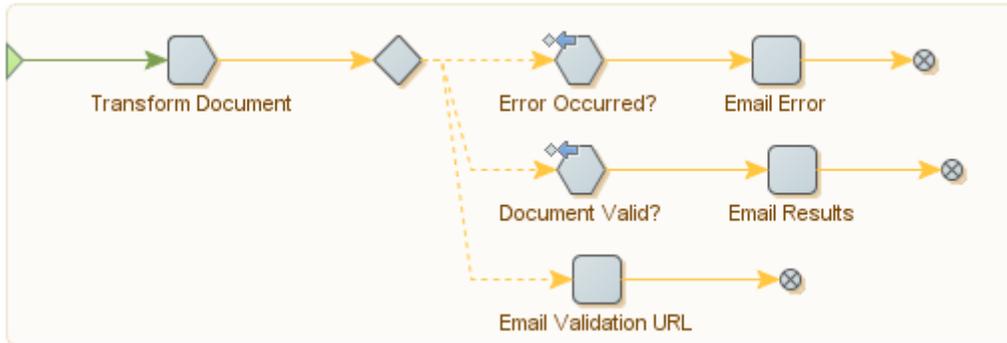
5. 2番目の分岐に [アクション ステップ] を追加し、[値判定] ステップにします。
 - a. [条件] セクションで、「**ValidationURL==""**」と入力します。直線型の引用符を使用します。

- b. [If] および [Do] を [条件が満たされていません] および [[エラー処理] 時に指定されたとおり] にそれぞれ設定します。
 - c. [エラー処理] タブの [Then] リストで [次のトライステップへ移動] を選択します。
 - d. ロボットで、[値判定] ステップの名前を「**Document Valid?**」に変更します。
6. [値返却] ステップをクリックして、[メール送信] ステップにします。
7. 2 番目の [メール送信] ステップを次のように設定します。
- a. [送信元アドレス] フィールドと [宛先アドレス] フィールドに、送信者と受信者の電子メール アドレスをそれぞれ入力します。
 - b. このステップの [メッセージ] フィールドに [エクスペッション] を設定し、以下のエクスペッションを追加します。
"VendorID=" + invoice.VendorID + "\n" + "InvoiceNumber=" + invoice.InvoiceNumber + "\n" + "InvoiceDate=" + invoice.InvoiceDate + "\n" + "Total=" + invoice.Total
 - c. [メッセージ タイプ] を [テキスト] に設定します。
 - d. [件名] フィールドに、次のようなテキストを入力します。"This is the result of the robot execution."
 - e. [サーバー] タブで、SMTP サーバーを設定し、変更を保存します。
 - f. ロボットで、ステップの名前を「**Email Results**」に変更します。



8. 3 番目の分岐に [アクション ステップ] を追加し、[メール送信] ステップにします。
9. 2 番目の [メール送信] ステップを次のように設定します。
- a. [送信元アドレス] フィールドと [宛先アドレス] フィールドに、送信者と受信者の電子メール アドレスをそれぞれ入力します。
 - b. このステップの [メッセージ] フィールドに [エクスペッション] を設定し、以下のエクスペッションを追加します。
>><<a href="<<+ValidationURL+>>">Click me!<<
 - c. [メッセージ タイプ] を [HTML] に設定します。
 - d. [件名] フィールドに、次のようなテキストを入力します。"Please validate this invoice."
 - e. [サーバー] タブで、SMTP サーバー設定を行います。
 - f. ロボットで、ステップの名前を「**Email Validation URL**」に変更し、変更内容を保存します。

これでロボットが使用できるようになり、以下のようになります。



i [メール送信] ステップはデザイン モードでは実行できないため、デバッグ モードで実行する必要があります。このモードに切り替えるには、ツールバーで [デバッグ]  をクリックし、[シングル ステップ]  をクリックします。

デザイン モードとデバッグ モードを同時に実行することもできません。

ドキュメントが有効な場合、ロボットは変換されたドキュメントからの次の値を電子メールで送信します: VendorID、Number、Date、Total。変換されたドキュメントが無効な場合、検証 URL を含む電子メールが人に送信され、人はその URL をクリックしてドキュメントを検証できます。変換プロセス中にエラーが発生した場合、エラー情報を含む電子メールが人に送信されます。

ロボットを Management Console にアップロードします。これでロボットを Design Studio でテストできるようになりました。

検証に必要なドキュメントはすべて以下の URL から利用可能です。

`http://127.0.0.1:50082/Validation.aspx`

この URL の **127.0.0.1** は Thin Client サーバーの IP アドレスであり、インストール場所に応じて異なります。

人による検証の手順については、次のセクションを参照してください。

変換されたドキュメントの検証

このトピックでは、変換された請求書を検証するために実行できる基本的なステップについて説明します。

ブラウザで検証 URL `http://127.0.0.1:50082/Validation.aspx` を開いた後、変換されたドキュメントを含むバッチを開きます。左側の [Classification Result] ペインに、Document Transformation 結果が表示されます。

i ドキュメント検証プロセスを迅速かつ容易に行うために、可能な場合は必要に応じてキーボードを使用することをお勧めします。たとえば、バッチを開く場合は Ctrl+O をバッチを、サスペンド (保存して閉じておく) する場合は Ctrl+S を押します。ショートカットを使用するには、キーを同時ではなく 1 つずつ連続して押します。

1. [ID Number] フィールドの横にあるフィールドが選択され、ここで ID を確認する必要があります。

ID Number	198000		<input type="text"/>
Company Code	3		

正しい ID が抽出されたことを確認するには、Enter を押します。

ID Number	198000		Confirmed
Company Code	3		

2. 次の無効である可能性があるフィールド [Invoice Number] (請求書番号) が選択されます。

- Invoice -

Invoice Number	67-90943
----------------	----------

ドキュメントの右上隅の数字を確認できます。赤くマークされ、検証が必要であることが通知されます。

Date: 06.11.2010
Invoice No: 67-90943

フィールドを検証し、正しい請求書番号が抽出されていることを確認するには、Enter を押します。

3. 次の無効である可能性があるフィールド [Invoice Date] (請求書日付) が表示されます。

Invoice Date	04.11.2010
--------------	------------

ドキュメントの右上隅に、実際の日付が 06.11.2011 と表示されます。抽出された日付は、トレーニングの目的で意図的に無効になっています。

日付を修正するには、次のうちいずれかの方法に従います。

- 自動フィールド入力を使用します。まず、Backspace キーを使用して、正しくない日付をクリアします。ドキュメントのゼロで始まるすべての数字を表示するには、「0」と入力します。下矢印キーを押し、「06.11.2010」を選択します。

Invoice Date

06.11.2010
06.11.2010
04.11.2010
0000

ドキュメントの請求書日付は緑でマークされています。

Date: 06.11.2010
Invoice No: 67-90943

Enter を押して、新しい日付を確認します。

i ドキュメントの検証時に、オンライン ラーニングを有効にして、同じようなドキュメントでのフィールド認識の確率を上げることができます。この機能は、請求書などのサンプルドキュメントのレイアウトの記憶に基づいています。自動フィールド入力を使用、ドキュメントに正しい値 (正しい請求書日付など) を手動で入力または選択することにより、ナレッジ ベースに貢献します。その結果、次回同様のドキュメントを表示する際に、請求書日付は自動的に正しいものになります。

この機能は、[プロジェクト設定] の [一般] タブで有効になっています。

- ドキュメントで、請求書日付をクリック、または投げ縄をかけて選択し、Enter を押します。

Date: 06.11.2010
Invoice No: 67-90943

- 次の無効である可能性があるフィールド [PO Number] (注文書番号) が表示されます。

PO Number
4500017282

抽出された数字が正しいことを確認します。注文書番号が正しいことを確認すると、無効とマーク付けされていた [Document Type] (ドキュメント タイプ) フィールドの値が自動的に確認されます。

- [Currency] (通貨) フィールドが選択されます。

Currency
GBP

抽出された通貨が正しいことを確認します。

すべての値を確認すると、バッチが有効であるという通知を受け取ります。これで通知を閉じることができます。抽出された値は、**ロボットが無効なドキュメントと有効なドキュメントを処理できるように設定** で設定したように、電子メールで人に送信されます。

これでチュートリアルが完了しました。