



Kofax RPA

ベストプラクティス ガイド

バージョン: 11.4.0

日付: 2022-11-18

KOFAX

© 2019–2022 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.....	5
第 1 章：ロボット ライフサイクル マネジメント.....	7
ベーシック セットアップ.....	7
ロボット ライフサイクル マネジメントを使用しない基本設定.....	7
ロボット ライフサイクル マネジメントを使用した基本的な設定.....	8
ロボット ライフサイクル マネジメントを使用した基本的な設定.....	9
プランチ選択の方針.....	9
ペア リポジトリの作成.....	13
Management Console の設定.....	14
同期の開始.....	15
マージさせる/変更を元に戻す.....	18
同期結果の確認.....	19
アクセス権と前提条件.....	19
第 2 章：Desktop Automation サービス セットアップ.....	21
複数の DAS セットアップへのラベルの使用.....	21
設定ファイルを使用した Desktop Automation サービスの自動化.....	24

はじめに

このガイドでは、Kofax RPA を使用したパフォーマンスの最適化を確実に実行するために推奨される方法とテクニックを説明します。

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/11.4.0-vcsft2fhaw/RPA.htm>

このガイドの他に、ドキュメントセットには次の項目が含まれています。

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 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 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 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 ロボット構築スタート 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。

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 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 Desktop Automation* サービス設定ガイド**

リモート コンピュータで Desktop Automation を使用するためには必要な Desktop Automation サービスを設定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 API に関する情報が含まれています。

Kofax RPA Integration API documentation (Kofax RPA 統合 API ドキュメント)

Kofax RPA へのプログラムでのアクセスを提供する Kofax RPA Java API および Kofax RPA .NET API についての情報が含まれています。Java API 文書は、オンラインおよびオフラインの Kofax RPA 文書から入手できますが、.NET API 文書はオフラインのみとなります。

i 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。

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 Base \(Kofax ナレッジベース\)](#)] 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 Base で情報を検索することをお勧めします。

[Kofax Knowledge Base] にアクセスするには、次の手順を実行します。

1. [Kofax Web サイト](#) のホーム ページに移動して、[サポート] を選択します。
2. [サポート] ページが表示されたら、[カスタマーサポート] > [ナレッジベース] を選択します。

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

[Kofax Knowledge Base] 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。
- [Kofax Knowledge Base] のホームページをスクロールして、製品ファミリーを見つけてください。目的の製品ファミリーネームをクリックして、関連記事の一覧を表示します。一部の製品ファミリーの場合は、関連記事を表示するために Kofax Portal (Kofax ポータル) の有効なログイン情報を入力する必要があります。

ナレッジベースのホーム ページから、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)。
ページ上部の [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)。
ページ上部の [Support (サポート)] リンクをクリックします。[Customer & Partner Portals Overview (カスタマーとパートナー ポータル概要)] が表示されたら、[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)。
ページ上部の [Support] リンクをクリックします。[Customer & Partner Portals Overview] が表示されたら、[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。
- [General Support (一般サポート)] セクションに移動して、[Support Details (サポートの詳細)] をクリックし適切なタブを選択します。

第1章

ロボットライフサイクルマネジメント

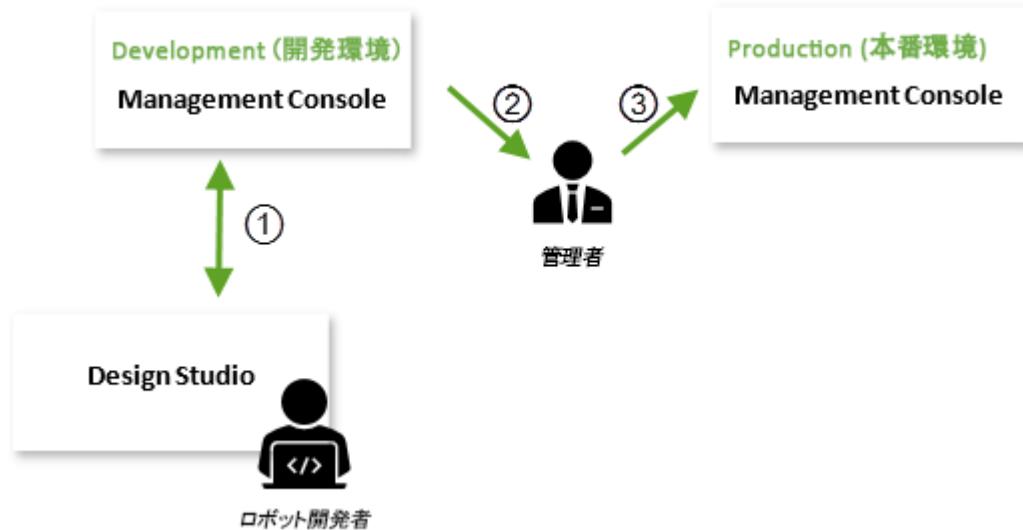
ロボットライフサイクルマネジメント機能を使用すると、Gitなどのバージョン管理システムでさまざまなタイプの作業オブジェクトを制御できます。この機能を使用すると、インストールに含まれる Kofax RPA Synchronizer を利用して、Management Console とリポジトリの間のファイルの状態を比較および同期できます。同期できるオブジェクトは、ロボット、タイプ、スニペット、リソース、スケジュール、OAuth です。

これらのベストプラクティスは、Git バージョン管理システムについて実用的な知識があることを前提に記述されています。

ベーシックセットアップ

ロボットライフサイクルマネジメントを使用しない基本設定

次の図に、ロボットライフサイクルマネジメントを使用しない Kofax RPA の基本的な推奨環境を示します。この設定は、開発用と本番用の 2 つの Management Console で構成されています。



この設定例では、ロボット開発者は開発用 Management Console に保存されているプロジェクトとの同期を確立し (1)、新しいファイルや設定を作成するか既存のものに変更を加えて、この Management Console とプロジェクトを再同期します。

管理者は、本番用のプロジェクトを検討するときに、このプロジェクトのバックアップを作成するか(2)、開発用 Management Console の特定のファイルや設定のバックアップを作成し、このバックアップを本番用 Management Console にアップロードします(3)。

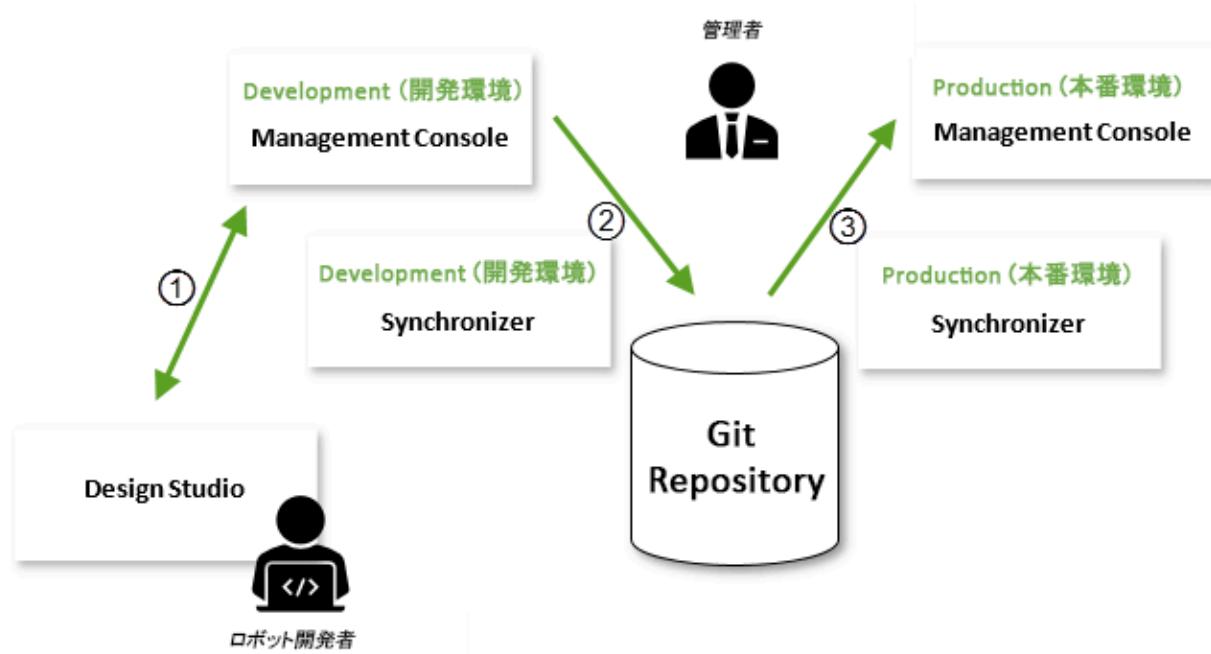
ロボットライフサイクル マネジメントを使用しない場合、この環境構成は次の機能を使用できません。

- ・プロジェクト内の各ファイルや設定の正確なバージョン履歴。
- ・現在本番稼働中のバージョンを特定し、変更の作成者、ファイルや設定の最終変更日、および変更について説明するメッセージを表示する機能。
- ・ファイルや設定の特定のバージョンを開発から本番に昇格させる機能。
- ・本番環境のファイルや設定のバージョンを以前のバージョンにすばやく戻す機能。
- ・開発環境の重大な変更を元に戻す機能。

これらのすべての機能を Kofax RPA 環境で使用できるように追加するには、次のトピックの手順に従って、ロボットライフサイクル マネジメントを使用した基本設定を構成します。

ロボットライフサイクル マネジメントを使用した基本的な設定

次の図に、ロボットライフサイクル マネジメントを使用した Kofax RPA の基本的な推奨環境を示します。この環境では、単一のリポジトリを共有する 2 つの Management Console を示します。1 つは開発用の Management Console で、もう 1 つは本番用の Management Console です。



この例では、ロボット開発者は開発用 Management Console に保存されたプロジェクトとの同期を確立します(1)。

ロボット開発者が開発用 Management Console 内のプロジェクトを更新すると、変更は共有リポジトリと自動的に同期し (2)、変更コミットとして指定したブランチにプッシュされます。プロダクションマネージャーは、本番環境で使用するブランチを指定することに責任を負います。

この時点で、変更を推進する方法は 2 つあります。

- 変更の結合、ベースの再設定、選択、本番用 Management Console で指定されたブランチへのプルを行う。
- 新しいバージョンに対して本番用 Management Console を単に指定する。

管理者が変更を承認すると、本番用 Synchronizer はリポジトリから変更を取得し (3)、本番用 Management Console にプッシュします。

次のトピックでは、この設定を段階的に再現します。

ロボット ライフサイクル マネジメントを使用した基本的な設定

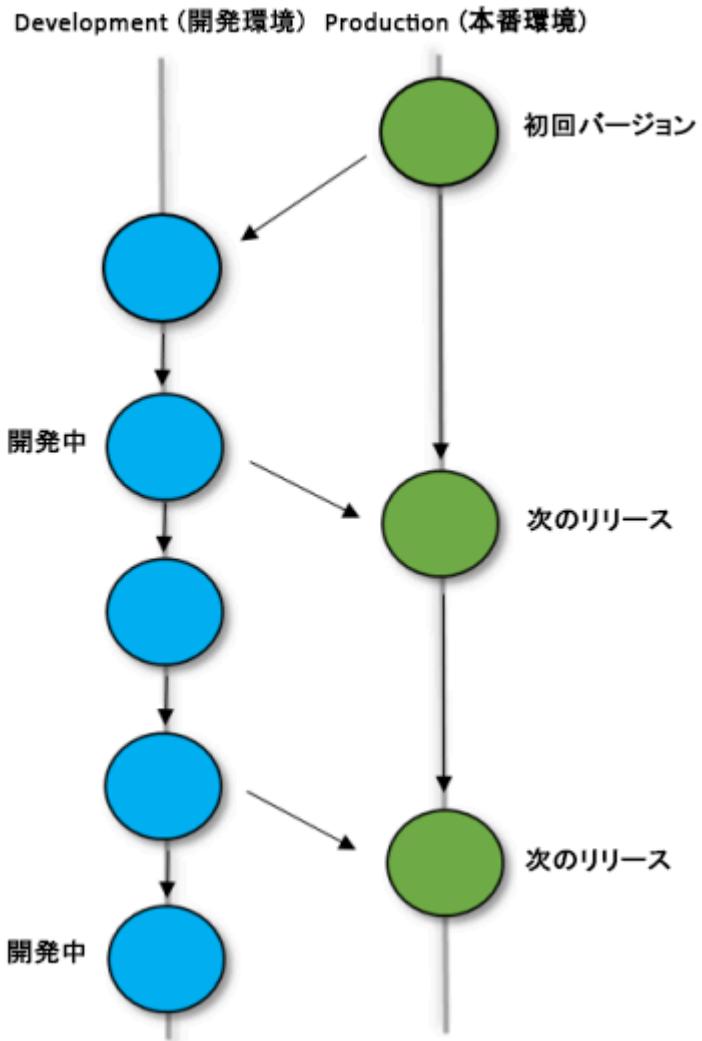
ブランチ選択の方針

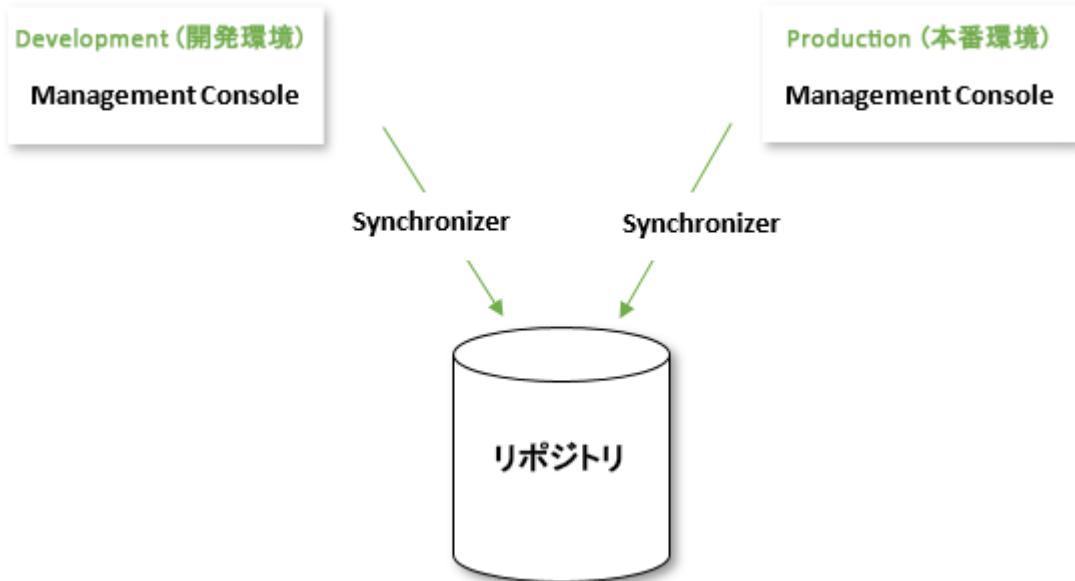
Git リポジトリを使用してファイルや設定を管理する場合は、さまざまな方法を使用できます。次のトピックに示す戦略と推奨事項に従うことをお勧めします。

推奨ブランチ戦略

この戦略に従って、マスター(本番)ブランチの先頭には、本番環境で実行されている現在のバージョンが常に含まれています。開発用などの別のブランチをマスター(本番)ブランチに新しく追加して、開発用 Management Console と新規作成した開発用ブランチを同期することができます。

次の図は、推奨の最小限のブランチ戦略と設定を示しています。青と緑の円は、それぞれ開発ブランチと本番ブランチの現在のバージョンを含む Git ヘッドです。





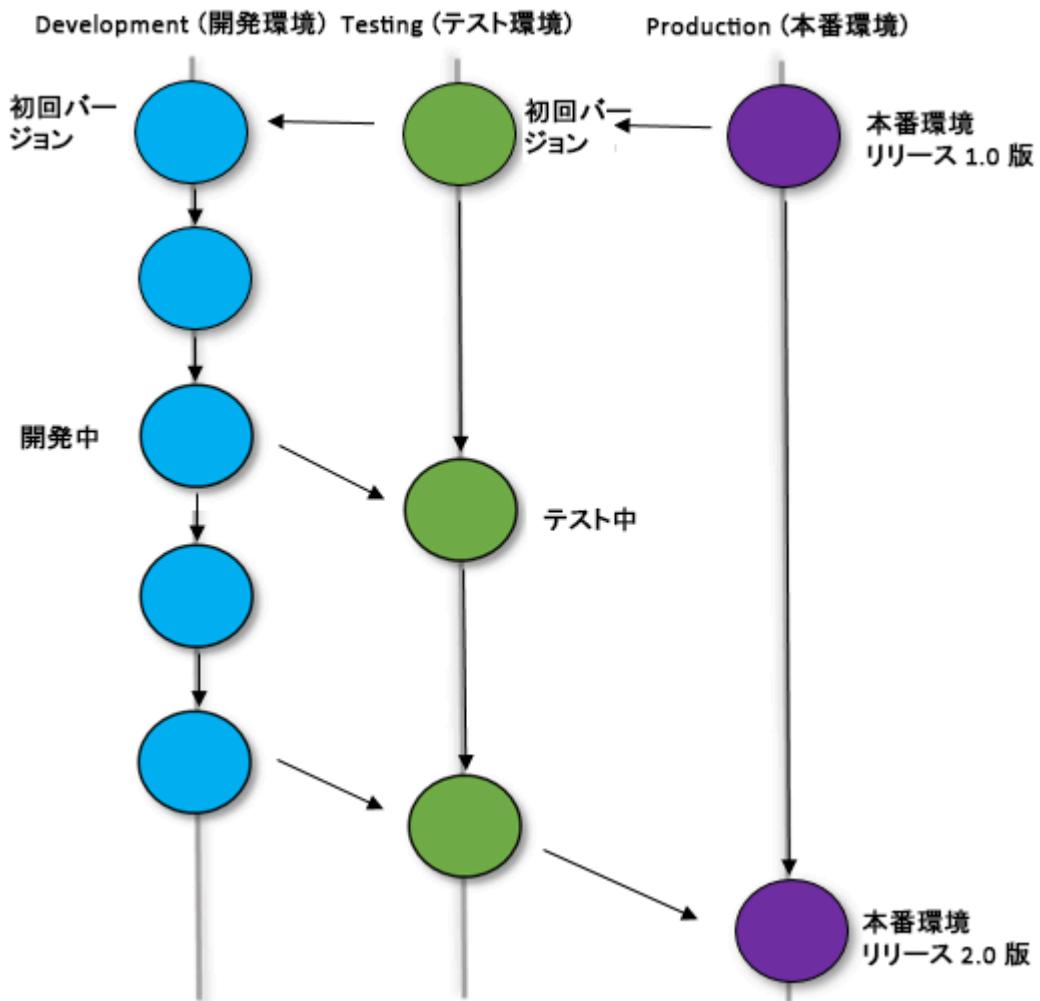
推奨 1: 本番環境では常にマスター ブランチを使用します。

推奨 2: マスター ブランチに結合する場合は、`--no-ff` フラグを使用します。このフラグを使用すると、現在のヘッドがマージしているコミット以前の変更であることが検出された場合に、Git が「早送り」方式で実行されないように防止されます。マージの正確な日付と時刻を追跡するように、本番ブランチ上のマージコミットを設定すると便利です。

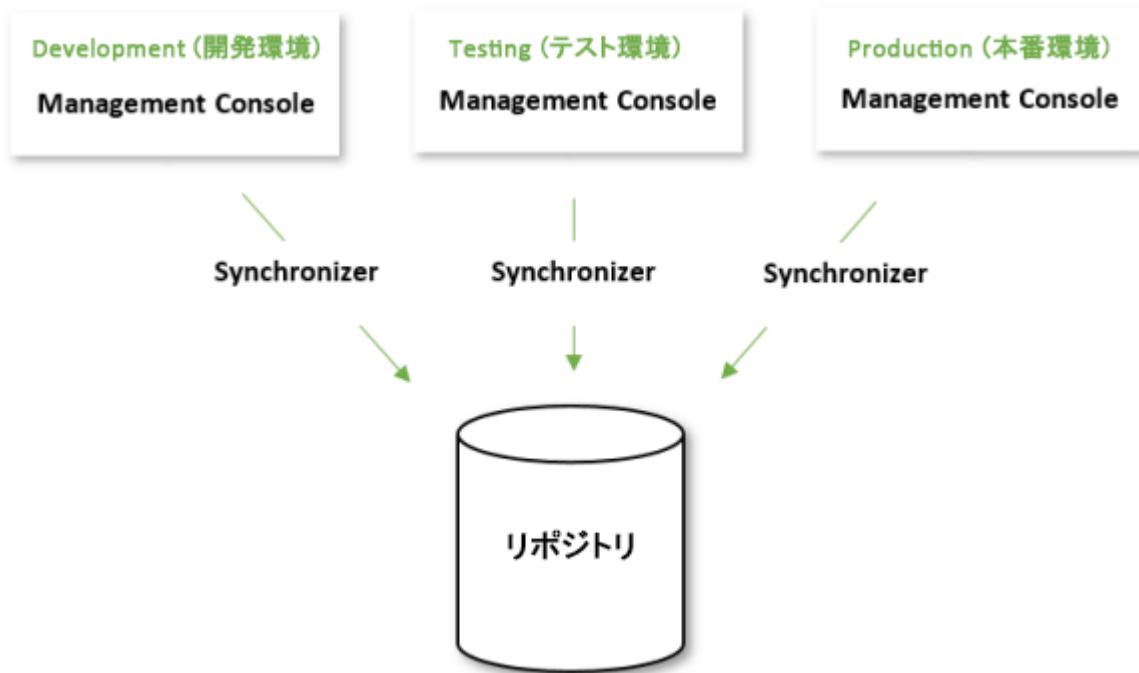
① 上記の戦略の詳細については、「[成功する Git ブランチモデル](#)」を参照してください。

その他のブランチ戦略

さらに複雑な設定が必要な場合は、前のトピックで示した最小限のブランチ戦略を拡張して、開発、テスト、本番の 3 つのブランチを持つ大規模な設定に対応させることができます。



この設定では、開発、テスト、および本番の 3 つの Management Console が、Synchronizer を介して Git リポジトリ内の 3 つのブランチと同期しています。



ベアリポジトリの作成

Synchronizer を起動して Management Console を同期するように設定する前に、Git をインストールし Git Bash を開いて Git リポジトリを初期化する必要があります。新しいベアリポジトリを作成する方法は、使用しているサードパーティツールによって異なります。

ベアリポジトリと開発ブランチを作成するには、次のコマンドを実行します。

```
# mkdir example.git
# cd example.git/
# git init --bare
Initialized empty Git repository in /gitrepos/example.git/
# cd ..
# git clone example.git
Cloning into 'example'...
warning: You appear to have cloned an empty repository.
done.
#cd example/
# git commit --allow-empty -m 'initial commit'
[master (root-commit) b96fdb8] initial commit
# git push origin
Everything up-to-date
# git checkout -b development
Switched to a new branch 'development'
# git push -u origin development:development
```

i または、Synchronizer を使用してペアリポジトリを自動的に作成します。Synchronizer を起動すると、ペアリポジトリがまだ作成されていない場合は指定した場所に自動的に作成され、初期コミットが空になります。ただし、Synchronizer がまだ存在しない場合は、ブランチが作成されません。

Management Console の設定

リポジトリを作成した後で、開発用と本番用の 2 つの Management Console を設定する必要があります。これらは、ファイルベース リポジトリ内の開発ブランチおよび本番 (マスター) ブランチと同期するために使用されます。

開発用 Management Console の設定

1. 開発用 Management Console を起動します。
2. メニューで [管理] > [プロジェクト] を選択し、リポジトリと同期するプロジェクトの : コンテンツメニューをクリックして、[編集] をクリックします。
3. 新しいダイアログ ボックスで、[リポジトリ] タブを選択します。
 - a. [URL] プロパティに、[前のトピック](#)で作成したリポジトリのパスを入力します。例: `/gitrepos/example.git`
 - b. [ブランチ] プロパティで、使用するブランチ名 (例: **development**) を入力します。
 - c. 上記の設定を有効にするには、[設定を有効化] を選択します。
 - d. [同期するオブジェクト] で、同期に含めるオブジェクトを[ロボット、タイプおよびスニペット] から選択します。

設定を有効化

URL
/gitrepos/example.git

ブランチ
development

読み取り専用

同期するオブジェクト

- スケジュール
- ロボット、タイプおよびスニペット
- リソース
- OAuth

- a. 変更を保存します。

本番用 Management Console の設定

1. 本番用 Management Console を起動します。
2. メニューで [管理] > [プロジェクト] を選択し、リポジトリと同期するプロジェクトの : コンテンツメニューをクリックして、[編集] をクリックします。
3. 新しいダイアログ ボックスで、[リポジトリ] タブを選択します。
 - a. [URL] プロパティに、[前のトピック](#)で作成したリポジトリのパスを入力します。例: `/gitrepos/example.git`
 - b. [ブランチ] プロパティで、使用するブランチ名 (**master**) を入力します。本番には、常にマスター ブランチを使用することをお勧めします。
 - c. リポジトリをオブジェクト変更の唯一のソースにするには、[読み取り専用] を選択します。本番用 Management Console 内の同期されたプロジェクトに属するオブジェクトが変更されないようにするには、このオプションを選択することをお勧めします。
 - d. 上記の設定を有効にするには、[設定を有効化] を選択します。
 - e. [同期するオブジェクト] で、同期に含めるオブジェクトを[ロボット、タイプおよびスニペット] から選択します。

設定を有効化

URL
/gitrepos/example.git

ブランチ
master

読み取り専用

同期するオブジェクト

- スケジュール
- ロボット、タイプおよびスニペット
- リソース
- OAuth

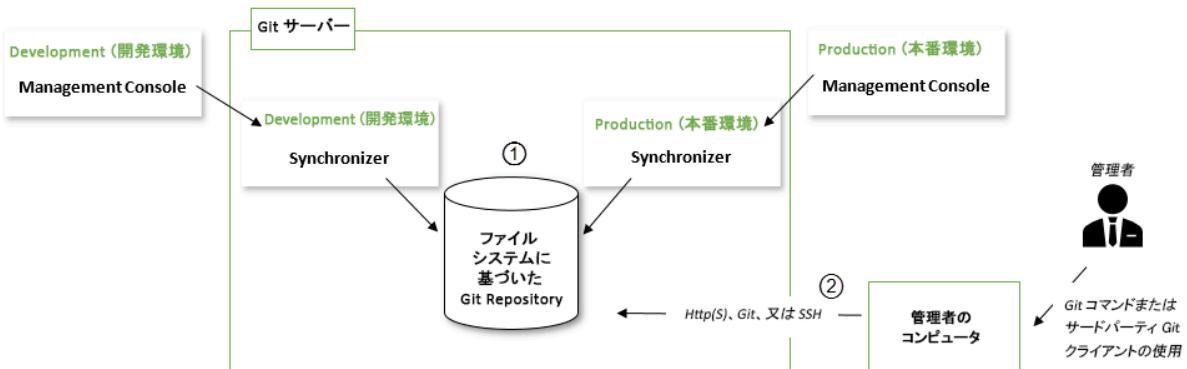
- f. 変更を保存します。

これで、Management Console の設定が完了し、同期を開始する準備が整いました。

同期の開始

ファイルベース Git リポジトリを使用する場合は、開発用 Management Console と本番用 Management Console の Synchronizer インスタンスから、リポジトリが配置されているファイルシステムに直接アクセスできる必要があります。

リポジトリが保存されているのと同じコンピュータで Synchronizer の両方のインスタンスを実行することをお勧めします (1)。この設定の場合、プロモーションマネージャーはサポートされている Git プロトコル標準 (2) の 1 つを使用してリポジトリにアクセスし、変更を本番環境にプッシュできます。



推奨 1: ファイルベースのペア リポジトリと同期する場合は、複数の Synchronizer を同じコンピュータで実行できます。

推奨 2: 必要に応じて、管理者は Git コマンドの代わりに、SourceTree などのサードパーティの Git クライアントを使用できます。

推奨 3: Docker を開発に使用する場合は、Synchronizer コンテナでボリュームを共有できます。

同期を開始する

Management Console への Synchronizer アクセスを構成します

2 つの Management Console と Git リポジトリを接続するには、次の手順を実行します。

1. **synchronizer.settings** ファイルで、開発 Synchronizer を開発 Management Console にポイントし、次の表のコマンドを使用して設定を構成します。
 - a. コマンドプロンプト ウィンドウで、**synchronizer.exe** ファイルに対して **-c** プロパティを指定して、デフォルトの **synchronizer.settings** ファイルを無視します。
 - b. 必要なその他の設定を指定します。

例:

```
Synchronizer.exe -c --mc-url http://127.0.0.1:8080/ManagementConsole --oauth-client-secret <insert-secret-string> --interval 10 --no-host-key false --private-key $USER_HOME \.ssh\id_rsa -s
```

コマンド	説明
-c,--command-line	command-line で指定した設定を使用し、設定ファイルは無視します。
-e,--environment	環境設定を使用し、設定ファイルを無視します。
-g,--generate-ssh-keys <argument>	SSH 認証用のキー ペアを生成し、指定されたフォルダに保存します。 例 : -g C:\Work\MyKeys

コマンド	説明
--mc_url <argument>	必須です。-c が指定されている場合にのみ使用します。プロトコルとポート番号を含む開発用または本番用 Management Console に接続するための URL を指定します。
--oauth-client-secret <argument>	必須です。-c が指定されている場合にのみ使用します。Management Console の OAuth サーバー セクションから Management Console を使用した認証シンクロナイザーにコピーされたプレーン テキストの OAuth クライアント シークレットを含みます。
--oauth-client-secret-file <argument>	必須です。Management Console の OAuth サーバー セクションから Management Console を使用した認証シンクロナイザーにコピーされた OAuth クライアント シークレットを含むファイルへのパス。
--interval <argument>	必須です。-c が指定されている場合にのみ使用します。同期の実行間隔を秒単位で設定します。0 以下の値または数値以外の値に設定した場合、Synchronizer は 1 回実行されて終了します。
--no-host-key <argument>	オプションです。-c が指定されている場合にのみ使用します。厳密な SSH ホスト キー チェックを無効にします。デフォルトは false です。
--private-key <argument>	必須です。-c が指定されている場合にのみ使用します。リモート リポジトリに接続するためのプライベート SSH キーを含むファイルへのパスを提供します。ローカル リポジトリに接続する場合、この属性は無視されますが、値を指定する必要があります。
-r,--reset-hard	バージョン情報をリセットし、ローカル キャッシュ全体をバージします。
-s,--save	synchronizer.settings ファイルに設定を保存して終了します。
-v,--version	バージョン情報を出力して終了します。
-h,--help	プロパティの説明を出力して終了します。

2. 本番用 Synchronizer および本番用 Management Console に前のステップを繰り返し、必要に応じてプロパティを変更します。

3. 設定を保存します。

Management Consoles と Git リポジトリが同期されました。オブジェクトを本番環境に昇格させ、Management Console プロジェクトと GIT リポジトリ間の同期を設定できるようになりました。

Management Console プロジェクトと github リポジトリ間の同期を設定する

Management Console プロジェクトと github リポジトリ間の同期を設定するには、次の手順に従います。

1. github リポジトリ Web サイトを開き、新しいアカウントを作成します。
2. コマンドプロンプト ウィンドウで、-g パラメータを指定して Synchronizer.exe ファイルを実行します。

例

```
Synchronizer.exe -c --mc-url http://127.0.0.1:8080/ManagementConsole --oauth-client-secret <insert-secret-string> -g $USER_HOME\.ssh\
```

これにより、公開鍵と秘密鍵の RSA ペアが生成されます。

3. github アカウントにログインします。[設定] メニューで [SSH および GPG キー] セクションを探し、生成されたキーペアの公開キーを入力して、新しい SSH キーを作成します。
4. github Web サイトで新しいプライベート リポジトリを作成し、リポジトリの SSH アドレス (git@githubrepository.com/username/repository_name.git など) をコピーします。
5. [Management Console] > [管理] > [プロジェクト] に移動して、プロジェクトを選択します。[編集] をクリックし、プロジェクトの編集 ウィンドウで [リポジトリ] を選択します。
 - a. [設定を有効化] をクリックして、リポジトリを設定するためのフィールドをアクティブ化します。
 - b. [URL] フィールドに、コピーした SSH アドレスを入力します。
 - c. [プランチ] フィールドに、使用するプランチの名前を入力します。
 - d. [読み取り専用] が選択されている場合、プロジェクト ファイルは消去され、github リポジトリ ファイルに置き換えられます。[読み取り専用] が選択されていない場合、github リポジトリ ファイルは消去され、プロジェクト ファイルに置き換えられます。読み取り専用モードでは、Management Console に含まれるすべてのオブジェクトがリポジトリから取得されます。同期を開始する前に、同期するオブジェクトが Management Console で空であることを確認してください。空でない場合、エラー メッセージが表示されます。
6. オブジェクトの内容を削除するには、同期するオブジェクトリストから必要なオブジェクトを選択し、[選択したオブジェクトを削除] をクリックします。重要なデータを失わないように、オブジェクトを削除するときは注意してください。
 - e. [同期するオブジェクト] については、同期に含めるオブジェクトを選択します。
 - f. [OK] をクリックして変更を保存します。

6. コマンドプロンプト ウィンドウで、`--private-key` パラメータを指定して `Synchronizer.exe` ファイルを実行します。

例

```
Synchronizer.exe -c --mc-url http://127.0.0.1:8080/ManagementConsole --oauth-client-secret <insert-secret-string> --interval 10 --no-host-key true --private-key $USER_HOME\.ssh\id_rsa
```

これで、github リポジトリが Management Console プロジェクトと同期されました。

マージさせる/変更を元に戻す

ロボット、タイプ、およびスニペットを開発用 Management Console と同期する操作は、アップロード機能を使用して Design Studio から実行するか、[ロボット] タブを使用して Management Console インターフェイスから実行できるようになりました。この時点で、本番用 Management Console を変更するには、ファイルや設定のバージョンを開発から本番にマージさせる必要があります。

次のコマンドを使用して、開発プランチから本番(マスター)プランチに結合します。

```
# cd example
# git checkout development
# git pull
```

```
# git checkout master
# git merge --no-ff development
# git push
```

変更を元に戻す必要がある場合は、開発ブランチでコミットを以前の状態に戻してから、元に戻したコミットを本番（マスター）ブランチにマージします。

同期結果の確認

同期結果を確認するには、それぞれのログファイルを参照してください。デフォルトでは、ログファイルは `home\AppData\Local\Kofax RPA\version\Logs` にあります。ログファイルの場所は、`home\AppData\Local\Kofax RPA\version\Configuration` にある `log4j2_synchronizer.properties` で設定できます。

同期が成功した場合、ログには同期されたプロジェクトの名前と同期のコミット ID が含まれます。同期は指定された間隔で実行され、変更が表示されたらオブジェクトの変更が Management Console とリポジトリの間で自動的に同期されます。

変更が競合している場合、Management Console ではリポジトリ オブジェクトのバージョンが優先されます。この場合、競合する Management Console 変更は無視されます。

i 現在、リポジトリ内の同期されたファイルの名前や場所を変更することはできません。Management Console 内でこれらのファイルを参照するときに、不正確になるためです。

正常な同期には以下が含まれます:

- コミット ID。Management Console に同期されたリポジトリ オブジェクトの変更のコミットを示します。
- タイムスタンプ。この時点までに Management Console で変更されたすべてのオブジェクトが、リポジトリと同期されます。

Management Console インターフェイスで、新しく追加されたオブジェクトの作成者、コミット メッセージ、オブジェクト リビジョン、および最終変更日を確認するには、それぞれのビューに次の列が追加されていることを確認します。[変更者]、[コミット メッセージ]、[リビジョン番号]、および [最終変更日]。

アクセス権と前提条件

デフォルトでは、**admin**、管理者、プロジェクト管理者、または **VCS** サービス ユーザー ロールを持つユーザーのみがロボット ライフサイクル マネジメントを使用できます。

これらのロールにはすべて Synchronizer を起動する権限が含まれていますが、**VCS** サービス ユーザーはこの目的のために特別に設計されています。このロールを持つユーザーは、プロジェクトの同期の設定などの変更を Management Console に加えることはできません。したがって、このロールは、職務にプロジェクト管理が含まれていない、主にバージョン管理システムとの同期に関連した作業に責任を負うユーザーに割り当てることができます。

新しいユーザーにロボット ライフサイクル マネジメントへのアクセスを許可するには、Management Console で、これらのロールのいずれかを持つグループにユーザーを追加する必要があります。ユー

ユーザー ロールの詳細については、『Kofax RPA のヘルプ』の「ユーザーおよびグループの管理」を参照してください。

Synchronizer が文字に特定の非標準制限があるファイルシステムで動作するため、ファイルや設定名にそれらの制限がある文字が使用されている場合、ファイルシステム自体がサポートされていても、それらのファイルや設定の同期中に問題が発生する可能性があります。同期を正常に行うようにするには、ファイルシステムで同期ファイルや設定名に使用される文字が許可されていることを確認してください。

第 2 章

Desktop Automation サービス セットアップ

ロボット  がリモート コンピュータ上の Windows デスクトップ アプリケーションを自動化する必要がある場合に、Desktop Automation サービスが必要となります。詳細については、『Kofax RPA Desktop Automation Service Configuration Guide』(Kofax RPA Desktop Automation サービス設定ガイド) を参照してください。

複数の Windows デスクトップを使用して、アプリケーションを実行することができます。これらのデスクトップは Management Console に報告するように管理されています。ロボットは Management Console と対話して、利用可能かつ適切な DAS を取得します。これは、ロボットの実行に特定の条件または機能が必要であるためです。ロボットを適切な DAS と一致させるには、ラベルを使用します。

以下のセクションでは、DAS を使用するためのいくつかのベスト プラクティスについて説明します。

- [複数の DAS セットアップへのラベルの使用](#)
- [設定ファイルを使用した Desktop Automation サービスの自動化](#)

複数の DAS セットアップへのラベルの使用

RPA のセットアップでは、さまざまなコンピュータに Desktop Automation サービス (DAS) の複数のインスタンスがインストールされている場合があります。

複数の DAS インスタンスを同時に Design Studio に接続するには、それらのインスタンス用に個別のデバイスを作成する必要があります。その後、ベーシック エンジン ロボット  の「ロボットを呼び出す」ステップの [アクション] タブにある [必要なデバイス] にリストされたことを確認してください。

DAS がデスクトップまたは仮想マシンにインストールされている場合、こうしたコンピュータには通常、広範なソフトウェア アプリケーションのセットがインストールされていません。ラベルの追加は、それぞれのコンピュータにインストールされたソフトウェアや、ロボット  で使用可能なソフトウェアを識別するために役立ちます。

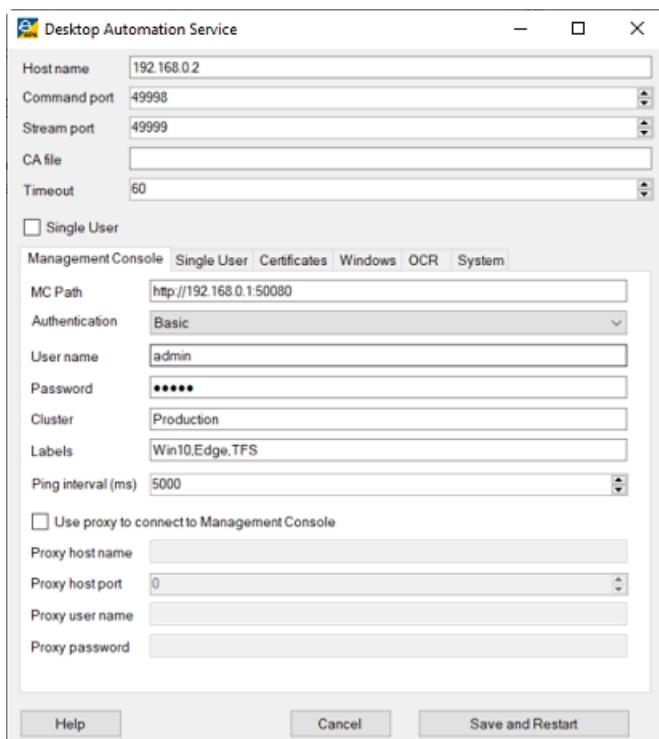
後ほど、ロボット  がコンピュータにインストールされている特定のビジネス アプリケーションを自動化する必要がある場合、システムはロボット  に必要なすべてのラベルを持つ利用可能な DAS を見つけます。

以下のようなシナリオが考えられます。

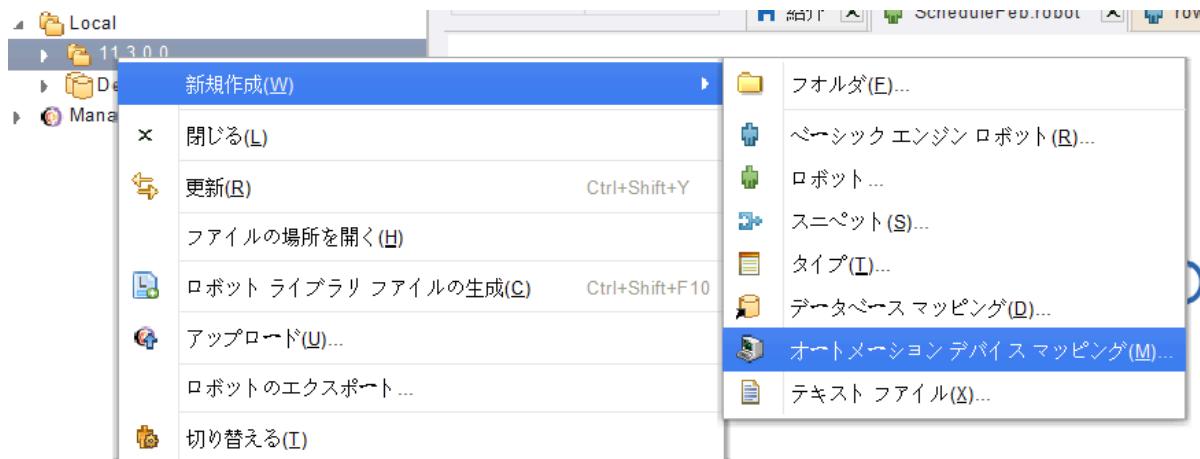
- 複数のセットアップにさまざまな DAS バージョンが使用されているが、自動化するビジネス アプリケーションは特定の DAS バージョンのみで機能する。
- さまざまなデスクトップがさまざまなプラットフォーム バージョン (Microsoft Windows) でセットアップされている可能性があり、それがロボットの  設定に影響を与えている場合がある。

以下の手順でラベルを割り当てます。

1. Desktop Automation サービスをスタートメニューから開始します。サービスが開始すると、そのステータスを通知領域のアイコンで確認できます。
2. 通知領域で Desktop Automation サービスアイコン  を右クリックし、[設定] を選択します。この操作により、Desktop Automation サービス ウィンドウが開きます。
3. [Management Console] タブの [ラベル] フィールドにラベルを入力します。以下の例では、Microsoft Windows 10、Microsoft Edge ブラウザ、および Team Foundation Server (TFS) アプリケーションがインストールされているコンピュータに DAS がインストールされています。ラベルはコンマで区切る必要があり、コンマと次のラベルの間にスペースを含めることはできません。組織内では、制限なしで任意のラベルを作成できます。



4. [保存して再起動] をクリックします。
5. [管理] > [デバイス] でデバイスが Management Console に登録されていることを確認します。
6. Design Studio を起動します。
7. [プロジェクト] リストで必要なプロジェクトを右クリックし、[新規] > [オートメーション デバイス マッピング] を選択します。



8. [新しいデバイスマッピング] 画面で新しいマッピングに名前を付け、[次へ] をクリックします。
9. [Management Console ベースのデバイスマッピング] が選択されていることを確認し、必須ラベルを入力します。以下の例では、ロボット  は、Microsoft Windows 10 がインストールされているコンピュータ上の TFS アプリケーションを自動化します。



10. 設定を完了し、ロボット  をこのデバイスにマッピングします。

! DAS を使用して 2 台のコンピュータでロボットを実行するには、[Management Console] で異なるラベルを持つ 2 つのデバイスマッピングを作成します。

! ロボットが Microsoft Windows ユーザーインターフェイスまたはローカライズされたアプリケーションと直接対話する必要がある場合は、プラットフォーム名および言語にラベルを使用することをお勧めします。これにより、ロボットは適切に設定された DAS インスタンスを選択できるようになります。

設定ファイルを使用した Desktop Automation サービスの自動化

DAS の展開を自動化する場合は、DAS のインストールに付属している元の設定ファイルを使用してください。他の製品バージョンのファイルは使用しないでください。

オートメーション デスクトップで `server.conf` ファイルを開きます。ファイルは `C:\>[ユーザー]>[ユーザー名]>[AppData]>[Local]>[Kofax RPA 11.4.0]` フォルダにあります。ユーザー名はサービスが実行されているユーザーの名前です。

DAS セットアップと一致するように、設定ファイルで次のパラメータを変更する必要があります。

- ホスト名
- MC パス
- ユーザー名
- パスワード
- (オプション) [ラベル](#)

ネットワーク セキュリティまたはその他の機能を設定するには、追加の設定が必要になる場合があります。