

Kofax TotalAgility

ベスト プラクティス ガイド

バージョン: 7.8.0

日付: 2020-07-06

KOFAX

© 2020 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

序文.....	7
関連ドキュメント.....	7
トレーニング.....	8
Kofax 製品のヘルプの入手.....	8
第 1 章: 展開トポロジ.....	10
Web/アプリー体型と Kofax TotalAgility Windows サービス.....	10
Web 用とアプリ用に別々の層に分割.....	11
Web/アプリー体型と分離型サーバーの Kofax TotalAgility Windows サービス実行.....	12
Web、コア サービス アプリ、および Kofax TotalAgility Windows サービス アプリ用に別々の層に分割.....	13
高可用性とスケーリング.....	14
高可用性.....	14
スケーリング.....	15
第 2 章: パフォーマンス.....	16
データベース.....	16
フォーム.....	18
プロセス.....	19
システム監視.....	19
キャプチャ クライアント.....	19
オブジェクト ID またはオブジェクト インデックスの収集をサポートする API.....	20
第 3 章: プッシュ アクティビティ処理時の Transformation Server インスタンスの構成.....	21
第 4 章: セキュリティ.....	22
保存データの保護.....	22
送信中のデータの保護.....	22
HTTPS 経由の送信データ.....	22
データベースで転送するデータ.....	22
権限.....	23
内部ユーザーと外部ユーザーへのサービス提供.....	23
Scan Client での悪意のあるファイルの アップロードに対する保護.....	23
サーバーの強化.....	24
悪意のあるデータベース接続と Web サービス URL に対する保護.....	25
第 5 章: データベース.....	26
サイジング.....	26
展開.....	26

メモリ.....	27
メンテナンス.....	27
回復機能.....	27
MSDTC.....	28
第 6 章: ソリューションの構築.....	29
システム設定.....	29
システム セッション ID.....	29
セッション タイムアウトおよびバッチ セッション タイムアウト.....	29
パスワード形式.....	30
パスワード ハッシュ アルゴリズム.....	30
パスワードを使用しないログオンの無効化.....	30
パスワードのリセット.....	31
複数のユーザー ログオンを許可.....	31
アカウント ロックアウト ポリシー.....	31
ビジネス カレンダー.....	32
電子メール アドレスの重複を許可する.....	32
監査ログへの書き込み.....	32
終了済みジョブのアーカイブ.....	32
レポート.....	33
フォームのキャッシュ.....	33
リフレッシュ間隔.....	34
スキル レベル.....	34
例外処理.....	34
保持ポリシー.....	35
チームベースの展開の構成管理.....	35
共有展開環境.....	35
スタンドアロン展開環境.....	36
新しいリリースを新しい環境にプロモートする.....	37
リソース.....	38
アクセス許可.....	38
アクティビティの割り当て.....	38
プロセスの構築.....	40
ケース対プロセス.....	40
フラグメントとプロセス.....	41
一般的なプラクティス.....	41
フォームの構築.....	46
フォームの使用の適合性.....	46
設計ガイドライン.....	46

キャプチャと非キャプチャの組み合わせ利用.....	47
フォームの再生成.....	47
フォーム/ナビゲーション/ヘッダーの再利用.....	48
セキュリティ.....	48
フォームの保守性.....	49
フォーム ロード イベント.....	49
フォームのテスト.....	50
作業キューから検証アクティビティをスキップしないようにする.....	50
ビジネス ルール.....	51
ビジネス ルールの保守性.....	51
ビジネス ルールのテスト.....	51
ジョブ アップグレード プログラム.....	51
データ管理.....	52
分散型のアップグレード可能性.....	52
第 7 章: トラブルシューティング.....	53
ビジネス プロセス管理.....	53
フォーム.....	53
プロセス.....	53
ビジネス ルール.....	54
パフォーマンス.....	54
全般.....	54
キャプチャ.....	55
Transformation Server.....	55
VRS.....	55
第 8 章: PDF の取り扱い.....	56
第 9 章: 抽出グループと分類グループの設計.....	57
分類グループおよび抽出グループ.....	57
共有プロジェクト.....	57
分類グループおよび抽出グループと共有プロジェクト.....	57
フィールド.....	58
強固な命名規則.....	58
シーケンスとフィールド グループ.....	58
ドキュメント フィールドの追加と削除.....	59
第 10 章: 検証とフォーマッタの実装.....	60
フォーマット設定.....	60
フィールド フォーマッタ.....	60
C# アクティビティと Visual Studio C# エディター.....	61
Validation.....	62

ビジネス ルールでのテーブル入力のキャプチャ.....	64
第 11 章: フォルダとドキュメントのロック.....	65
オブジェクトの削除.....	66
フォルダまたはドキュメントの入力変数.....	66
Web Capture コントロール.....	67
第 12 章: キャプチャ クライアント イメージ ビューアの画質.....	68
第 13 章: オンライン学習.....	69
オンライン学習システム タスク.....	69
トレーニング ドキュメントをインポートする間隔.....	69
第 14 章: パッケージの自動エクスポートおよび自動インポート.....	71
第 15 章: TotalAgility アプリケーションのパフォーマンスの監視.....	72
第 16 章: TotalAgility リンク済みサーバーの監視.....	76

序文

このガイドでは、パフォーマンス、コスト、メンテナンス、可用性、およびセキュリティを改善するために、TotalAgility の使用中に従う必要がある推奨ベスト プラクティスについて説明します。

関連ドキュメント

Kofax TotalAgility の製品ドキュメント セットは、次の場所で入手できます。

<https://docshield.kofax.com/Portal/Products/KTA/7.8.0-dpm5ap0jk8/KTA.htm>

このガイドのほか、ドキュメント セットには次の項目が含まれています。

- Kofax TotalAgility 前提条件ガイド: TotalAgility をインストールするためのシステム要件、前提条件ユーティリティの実行手順、およびさまざまなインストール タイプのソフトウェア チェックリストを説明します。
- Kofax TotalAgility インストール ガイド: TotalAgility をインストールし構成する方法について説明します。
- Kofax TotalAgility 統合サーバー インストール ガイド: Kofax 統合サーバーをインストールし、他の製品と統合する方法について説明します。
- Kofax TotalAgility オンプレミス マルチテナント インストール ガイド: オンプレミス マルチテナント システムのインストールおよび構成方法について説明します。
- Kofax TotalAgility 構成ユーティリティ ガイド: 構成ユーティリティを使用して、さまざまなタイプのインストールおよび展開の各種構成ファイルの設定を更新する方法について説明します。
- Kofax TotalAgility 管理者ガイド: TotalAgility インストールの構成および保守に関する情報を管理者に提供します。
- Kofax TotalAgility アーキテクチャ ガイド: オンプレミス、オンプレミス マルチテナンシー、Azure 環境のさまざまな展開を含む、TotalAgility アーキテクチャの概要について説明します。
- Kofax TotalAgility 機能ガイド: TotalAgility の機能概要を説明します。
- Kofax TotalAgility 移行ガイド: 異なるバージョンからの TotalAgility アップグレードおよびアップグレード後の構成に関する情報を説明します。
- Kofax TotalAgility のヘルプ: TotalAgility を使用してビジネス ジョブとケースを設計し、リソースを割り当て、フォームを作成し、外部アプリケーションと統合するなどの詳細を説明します。TotalAgility アプリケーションから [ヘルプ] ボタンをクリックすることで、ヘルプにアクセスできます。
- Kofax TotalAgility Workspace のヘルプ: Workspace を使用してアクティビティ、ジョブ、およびリソースを管理する方法について説明します。TotalAgility Workspace からヘルプにアクセスするには、[ヘルプ] ボタンをクリックします。
- Kofax TotalAgility オンプレミス マルチテナント システムのヘルプ: TotalAgility オンプレミス マルチテナント システムを使用してテナントを作成および管理する方法について説明します。
- Kofax TotalAgility Web Capture コントロールヘルプ: Web Capture コントロールを使用した、複数ページのドキュメントの作成、新しいフォルダでの新しいドキュメントの作成、誤ってスキャンされた

ページの削除などに関する詳細を提供します。また、Web Capture コントロール ツールバーで使用可能なボタンについても説明します。

- Kofax Analytics for TotalAgility 製品機能ガイド: ワークフローを通過するデータの追跡、プロセスとリソースの有効性の分析、ビジネス上の問題への対処に役立つダッシュボードの概要について説明します。
- Kofax TotalAgility テーブル: Kofax Analytics for TotalAgility で使用される Kofax TotalAgility テーブルおよびフィールドについて説明します。
- Kofax 製品からの移行ガイド: TotalAgility ファイルと Kofax 変換モジュール プロジェクトの TotalAgility への移行に関する情報を説明します。

トレーニング

Kofax では、お客様に Kofax TotalAgility ソリューションを最大限活用していただくことを目指して、教室でのトレーニングとコンピュータ ベースのトレーニングの両方を提供しています。利用可能なトレーニングの種類とスケジュールの詳細については、Kofax の Web サイト (www.Kofax.com) を参照してください。

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 (ナレッジ ベース) で情報を検索することをお勧めします。

[[Kofax Knowledge Base](#)] (Kofax ナレッジ ベース) を参照するには、[Kofax Web サイト](#) にアクセスして、ホームページで [サポート] を選択してください。

注 Kofax Knowledge Base (Kofax ナレッジ ベース) 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 (Kofax ナレッジ ベース) 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](#)] (Kofax ナレッジ ベース) 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ます。目的の製品ファミリー名をクリックして、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。
- [[Kofax Customer Portal](#)] (Kofax カスタマー ポータル) へのアクセス (資格のあるカスタマー向け)
ページ上部にある [**Customer Support**] (カスタマー サポート) リンクをクリックしてから、[**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 (資格のあるパートナー向け)
ページ上部にある [**Partner Support**] (パートナー サポート) リンクをクリックしてから、[**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) セクションまでスクロールして、**[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

展開トポロジ

この章では、最も一般的な展開オプションのメリットとデメリットについて説明します。

選択された展開オプションは、通常次のような要件により完全に決定されます。

- 良いパフォーマンス
- 低価格
- メンテナンスのしやすさ
- 単一障害点の防止、高可用性のサポート
- 適切なセキュリティ

Transformation Designer およびレポート サーバーを、それぞれ独自の専用サーバー上にインストールすることをお勧めします。どちらのアプリケーションも非常に集中的な CPU 動作を行います。独自のサーバーにインストールすると、パフォーマンスが向上します。

利用可能な展開トポロジは次のとおりです。

- Web アプリと Kofax TotalAgility Windows サービスの統合
- Web 用とアプリ用に別々の層に分割
- サービスを無効にした Web アプリと Kofax TotalAgility Windows サービスを実行する別のアプリケーション サーバーの統合
- Web、コア サービス アプリ、および Kofax TotalAgility Windows サービス アプリ用に別々の層に分割

展開トポロジでは、Transformation Designer とレポート サーバーがそれぞれ独自のサーバーにインストールされることを想定しています。

これらのトポロジでは、Kofax TotalAgility Windows サービスは次のとおりです。

- Core Worker サービス
- (インポート) Message Connector
- エクスポート ワーカー
- Streaming サービス

Web/アプリ一体型と Kofax TotalAgility Windows サービス

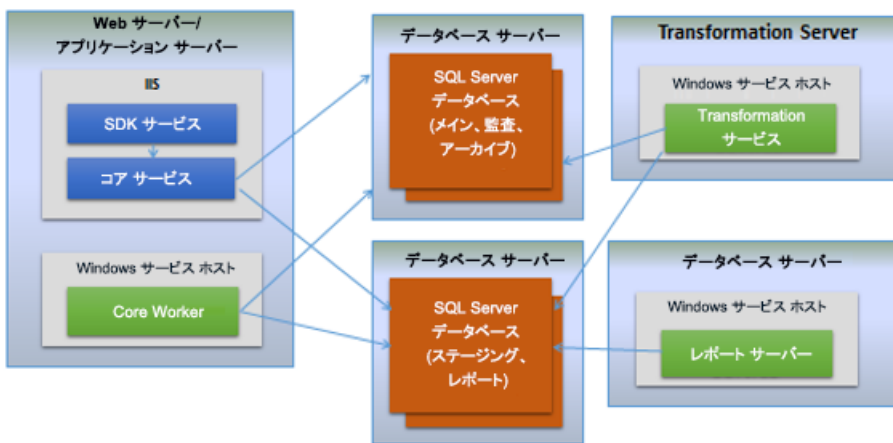
これは、公開された SDK サービス、コア Web サービス (SDK によって呼び出されるもの)、および Kofax TotalAgility Windows サービスがすべて同じサーバーにインストールされる最も基本的な展開タイプです。実行されている非キャプチャの自動アクティビティの量が少ない場合、この展開の条件を満たしています。

メリット:

- SDK からコア サービスへの呼び出しはメモリ内で生じるため、SDK への呼び出しが最も効率的です。

デメリット:

- Web サーバーはデータベースと直接通信します。
- 同じサーバー上に Web サーバーとアプリケーション サーバーがあると、追加のセキュリティ レイヤーを適用できる分割 Web とアプリケーションのように安全ではありません。
- Web サーバーには、バックグラウンド タスクを処理するコンポーネントが含まれているため、Web 要求の処理専用ではありません。



Web 用とアプリ用に別々の層に分割

この展開では、Web 層には公開された SDK Web サービスが含まれますが、Kofax TotalAgility Windows サービスは含まれません。アプリ層には、コア Web サービスと Kofax TotalAgility Windows サービスが含まれます。

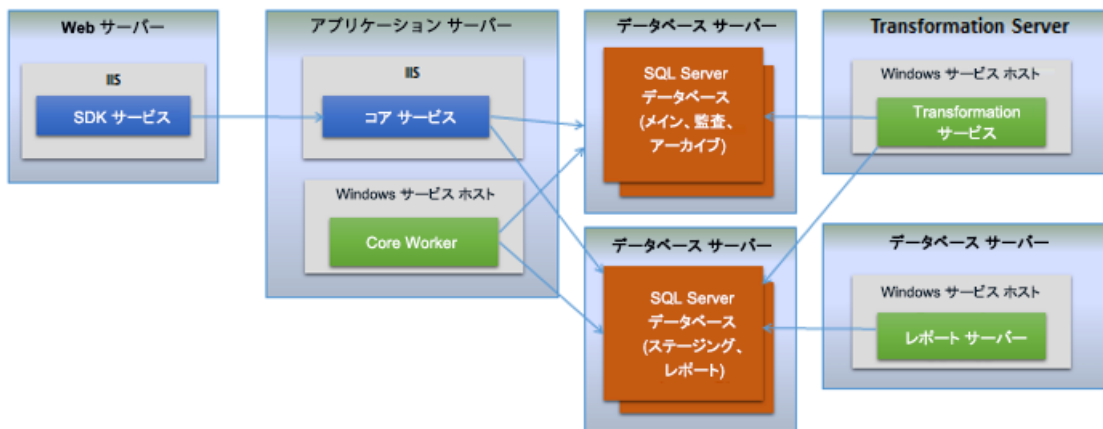
Web 層はデータベースに直接接続しないため、追加のセキュリティ レイヤーを提供します。コア Web サービスはデータベースと通信します。追加のセキュリティはパフォーマンスに影響し、この展開での SDK API の呼び出しでは、コア サービスへのコンピュータ間呼び出しが必要なため、Web/アプリ型体の展開よりも遅くなります。

メリット:

- Web サーバーはデータベースではなく、アプリケーション サーバーと通信します。つまり、追加のセキュリティ レイヤーをアプリケーション サーバーに適用できます。

デメリット:

- コア サービスは SDK サービスが存在するサーバーとは別のサーバーに存在するため、SDK への API 呼び出し時には、SDK からコア サービスにコンピュータ間呼び出しを行う必要があります。したがって、SDK への呼び出しは、[Web/アプリ型体](#)と [Kofax TotalAgility Windows サービス](#)の展開時のように効率的ではありません。



Web/アプリ体型と分離型サーバーの Kofax TotalAgility Windows サービス実行

Web/アプリ体型で Kofax TotalAgility がデフォルトでインストールするとおり、この展開では、Web サーバーには公開された SDK、コア Web サービス、および Kofax TotalAgility Windows サービスが含まれます。

Web/アプリ体型と Kofax TotalAgility Windows サービスの展開時と比較すると、この展開では Web サーバーが Kofax TotalAgility Windows サービスで生じる CPU の負荷から解放され、SDK の要求のパフォーマンスを向上できます。同様に、Kofax TotalAgility Windows サービスには SDK API の要求がロードされていないため、より効率的に実行できます。

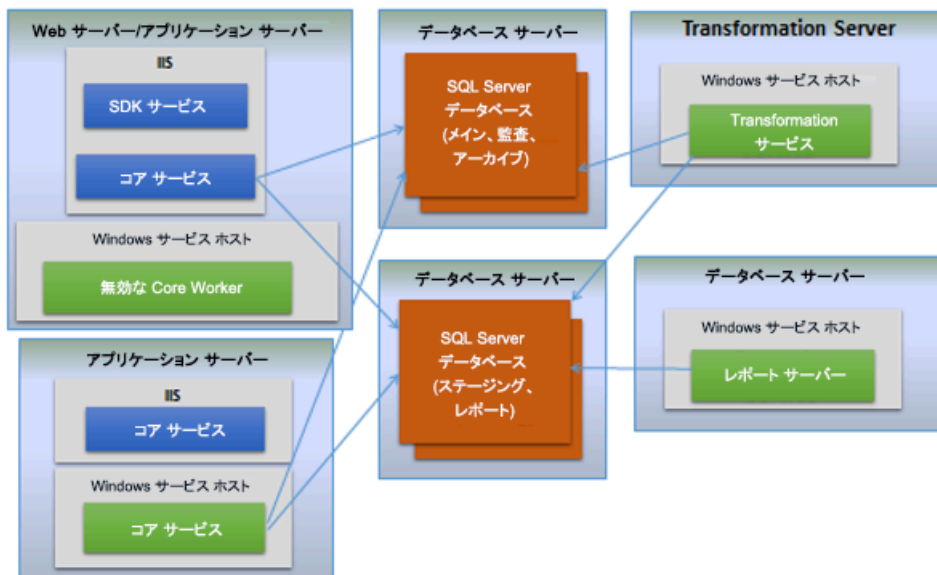
Web 用とアプリ用に別々の層に分割の展開時と比較して、この展開では SDK の要求のパフォーマンスが向上します。

メリット:

- SDK からコア サービスへの呼び出しはメモリ内で生じるため、SDK への呼び出しが最も効率的です。
- 専用アプリケーション サーバーは、バックグラウンド タスクの処理を担当します。

デメリット:

- Web サーバーはデータベースと直接通信します。



Web、コア サービス アプリ、および Kofax TotalAgility Windows サービス アプリ用に別々の層に分割

この展開と [Web アプリ](#)と [Kofax TotalAgility Windows サービス](#)を実行する別のアプリケーション サーバーの統合の展開時の主な違いは、そのコア Web サービスに独自の専用サーバーがあることです。この展開では、アプリ層の CPU 負荷が Web サーバーとアプリケーション サーバーに分散されます。Web サーバーは Web 要求を処理し、アプリケーション サーバーはシステム ワーカー タスクおよび非システム ワーカー タスクのバックグラウンド処理を処理します。

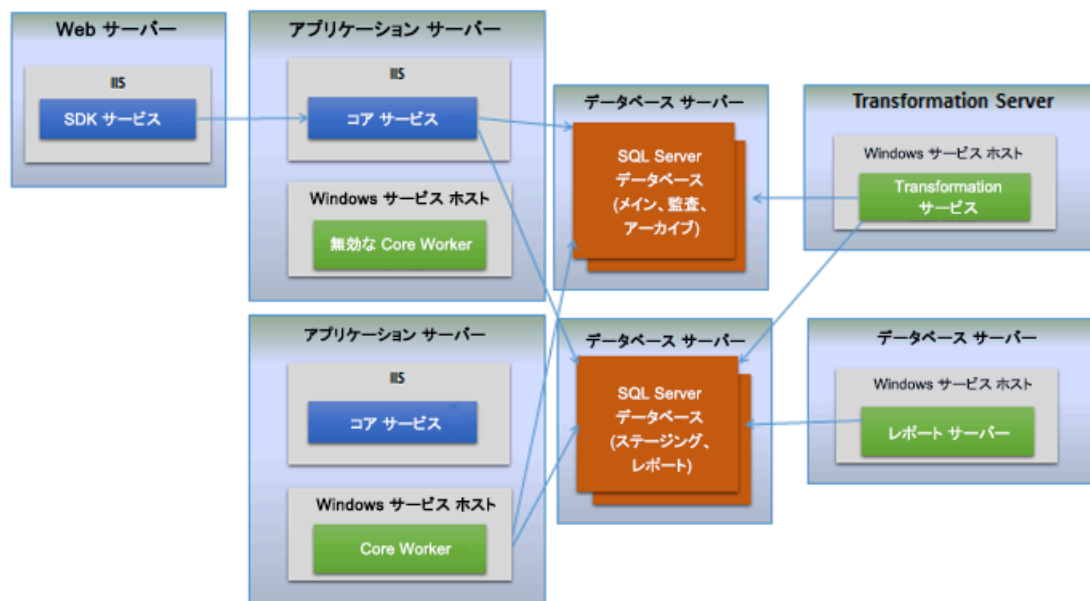
メリット:

- Web サーバーは、データベースではなくコア サービス アプリケーション サーバーと通信するため、セキュリティが向上します。
- Web 要求を受信するための専用のアプリケーション サーバー (コア サービスを含む)。
- バックグラウンド タスクを処理するための専用のアプリケーション サーバー (Kofax TotalAgility Windows サービスを含む)。

デメリット:

- コア サービスは SDK サービスがあるサーバーとは別のサーバー上に存在するため、SDK への呼び出しは、[Web /アプリ](#)と [Kofax TotalAgility Windows サービス](#)の展開時のように効率的ではありません

せん。したがって、SDK への API 呼び出しには、SDK からコア サービスへのコンピュータ間呼び出しが必要です。



高可用性とスケーリング

ソリューションを展開するとき (Kofax TotalAgility 以外も含む)、次のことに注意する必要があります。

- 単一障害点: 単一の障害が原因でアプリケーション全体の動作が停止する可能性があるアーキテクチャ内の場所。
- スケーラビリティ: 負荷の増加。

『Kofax TotalAgility アーキテクチャ ガイド』の「スケーリングと回復力」セクションも参照してください。

高可用性

高可用性を確保するために、次のことをお勧めします。

- 複数のサーバーにまたがって Kofax TotalAgility サービスをインストール、それらを水平にスケーリングします。このアプローチにより、複数のアプリサーバーに負荷が分散され、パフォーマンスが向上します。

注 すべての Kofax TotalAgility Windows サービスを単一のサーバーで実行する必要はありません。一部を無効にして、別のアプリケーションサーバー (または回復用サーバー) で実行するように設定できます。

- レポートサーバーおよび Transformation Server をそれぞれの専用サーバー上にインストールします。
- 高可用性 Web ファーム構成で複数の Web サーバーの負荷を分散します。このアプローチにより、Web 層もロード バランサーも単一障害点になりません。

- Web 層とアプリ層が分割され、ロード バランサーが Web サーバーとアプリ サーバーの間に位置する展開で、アプリ サーバーの負荷を分散します。
- データベースのクラスタリング、ミラーリング、またはレプリケーションを導入して、単一障害点を防ぎます。
- バックアップ ライセンス サーバーを使用して、障害から保護します。

スケーリング

増加する負荷要求に対応するために、水平または垂直スケーリングによりパフォーマンスを改善します。

垂直スケーリング: サーバーのメモリやコアなどを増設し、増加する負荷にハードウェアが対応できるようにします。

水平スケーリング: 複数のアプリケーション サーバーで同じソフトウェアを実行します。つまり、複数のアプリケーション サーバーで Kofax TotalAgility Windows サービスを実行します。Web サーバーの負荷分散は、Web サイトのパフォーマンスの向上に役立ちます。

さまざまなトポロジの利点と欠点については、「[可能な展開トポロジ](#)」を参照してください。

第 2 章

パフォーマンス

この章では、データベース、フォーム、作業処理、およびその他の領域でパフォーマンスを改善するための推奨ベスト プラクティスについて説明します。

データベース

このセクションでは、データベースに関するベスト プラクティスについて説明します。

- 終了済みジョブをアーカイブ テーブルに移動するか、ジョブが完了したら (または定期的に) パージします。このアプローチにより、ライブ データベース テーブルのサイズが最小化され、継続的にサイズが増加することがなくなります。
- 監査ログは、必要ない場合はオフにします。これにより、パフォーマンスが向上し、データベースのサイズが小さくなります。
- データベースの成長が最初から管理されるように、システム レベルとプロセス レベルの両方で保持ポリシーを定義します。
- 現在サポートされているすべてのバージョンの SQL Server は、オンラインでのインデックスのメンテナンスが可能です。この機能を利用するかどうかはユーザーが決定しますが、TotalAgility のパフォーマンスに影響を与える次の属性を考慮してください。
 1. 再構築を完了するために、サーバー リソースの使用率が増加します (メモリ、ディスク IO)。余分な読み取りと書き込みは、SQL Server がインデックス再構築の初期段階で作成するインデックスの 2 番目のコピーに起因します。
 2. クラスター化インデックスは、再構築が完了するまでインデックスとデータの両方のコピーを複製し、同時に起こるテーブル アクティビティのパフォーマンスに影響を与えます。非クラスター化インデックスは、リソースの使用率にわずかに影響します。
 3. REDO エントリの頻度が高いと、トランザクション ログは大幅にサイズが増加します。
 4. 準備フェーズおよびビルド フェーズ中のロックを強化して、インデックスの再構築中に他のプロセスがオブジェクトの排他ロックを取得しないようにします。
 5. 最後に取得したロックはスキーマ変更ロック (Sch-M ロック) であり、テーブルへの他のすべての同時アクセスをブロックします (古いインデックスが削除され、メタデータが更新されます)。

- SQL Server の並列処理を使用する場合、SQL Server の [プロパティ] > [詳細] で、次のプロパティを設定します。
- 並行処理 で、Cost Threshold for Parallelism を 35 に設定します。

サーバーのプロパティ - HV-JAP1-DEV

ページの選択

- 全般
- メモリ
- プロセッサ
- セキュリティ
- 接続
- データベースの設定
- 詳細設定**
- 権限

接続

サーバー:
HV-JAP1-DEV

接続:
sa

[接続のプロパティの表示](#)

進行状況

準備完了

スクリプト ヘルプ

FILESTREAM	
FILESTREAM アクセス レベル	無効
FILESTREAM 共有名	MSSQLSERVER
その他	
2 桁表記の年の基準になる年	2049
SQL Server の優先度を上げます	False
Windows ファイバーを使用する (簡易ブーリン)	False
アドホックワークロードの最適化	False
カーソルのしきい値	-1
スタートアップ プロシージャのスキャン	False
テキストレプリケーションの最大サイズ	65536
トリガーから他のトリガーの起動を許可する	True
フルテキストのアップグレード オプション	インポート
ブロック対象プロセスのしきい値	0
既定のフルテキスト言語	1033
既定の言語	English
ネットワーク	
ネットワーク パケット サイズ	4096
リモート ログイン タイムアウト	10
並列処理	
クエリの待機	-1
ロック	0
並列処理のコストしきい値	5
並列処理の最大限度	0

2 桁表記の年の基準になる年
[2 桁表記の年の基準になる年] オプションを使用して、2 桁の年を 4 桁の西暦年として解釈する場合に、基準になる年を 1753 ~ 9999 の範囲の整数で指定します。

構成した値(C) 実行中の値(R)

OK キャンセル

- SQL Server 2016 を使用する場合、各 TotalAgility データベースの [プロパティ] > [オプション] で次のプロパティを設定します。
- 自動 で、Auto Update Statistics Asynchronously を True に設定します。

データベースのプロパティ - TotalAgility

ページの選択

- 全般
- File
- ファイルグループ
- オプション
- 変更の追跡
- 権限
- 拡張プロパティ
- ミラーリング
- トランザクション ログの配布
- クエリストア

接続

サーバー:
HV-JAP1-DEV

接続:
sa

🔗 [接続のプロパティの表示](#)

進行状況

🔄 準備完了

照合順序(C): SQL_Latin1_General_CP1_CI_AS

復旧モデル(M): 完全

互換性レベル(L): SQL Server 2017 (140)

包含の種類(I): なし

その他のオプション(O):

FILESTREAM

FILESTREAM ディレクトリ名	
FILESTREAM 非トランザクション アクセス	Off

Service Broker

Broker が有効	True
Broker の優先度の許可	False
Service Broker 識別子	41fab258-8392-4baa-b51f-107deae90ab6

カーソル

コミットでカーソルを開じる	False
既定のカーソル	GLOBAL

その他

ANSI NULL 既定値	False
ANSI NULL 有効	False
ANSI PADDING 有効	False
ANSI 警告有効	False
Is Read Committed Snapshot On	True
NULL との連結で NULL を使用	False
VarDecimal ストレージ形式有効	True
スナップショット分離を許可	False
パラメーター化	単純

FILESTREAM ディレクトリ名

OK キャンセル

- SQL Server または各 TotalAgility データベースの最大並列処理回数 (MAXDOP) を、データベース サーバー上のコア数/ CPU 数に関する Microsoft の推奨に基づいて設定します。

フォーム

[フォームのキャッシュ](#)を使用するかどうかを決めて、使用する場合はサイズを決めます。定期的に変更されるフォームのパフォーマンスと、これらのフォームを保存するためのクライアント マシンのメモリとのトレードオフを考慮してください。デフォルトのキャッシュ サイズは適切に設定されています。

サイズの大きい作業キュー呼び出しやジョブ検索を行わないでください。50 行を超えるクエリが返されると、パフォーマンスに影響します。

プロセス変数で検索しないでください。この方法は古い方法であり、後方互換性のためだけに API で利用可能になっています。

プロセス

すべてのステップが自動の場合、同期プロセスを使用します。システムはこれらのプロセスを .NET コードに変換します。これにより、サーバーの負荷が大幅に削減されます。

デモンストレーションなどで、小さいボリュームでのパフォーマンスを検討するときは、パフォーマンスの問題を引き起こす可能性のあるシステムの動作に注意してください。例:

- 20 の自動アクティビティで非同期マップを実行する場合、各アクティビティの実行時間は数ミリ秒ほどですが、それらの間隔は 1 秒以上 (デフォルトの Core Worker ポーリング間隔) のため、プロセスマップの実行には 20 秒以上かかることになります。
- 自動キャプチャ アクティビティを実行する場合、Transformation Server はアクティビティがない場合に 30 秒間スリープするため、マップの実行が遅くなります。

システム監視

アプリケーション監視ソフトウェア Kofax Monitor を使用して、TotalAgility のサービス レベル メトリック、履歴パフォーマンス データ、およびリアルタイム処理ステータスを取得します。

このソフトウェアは、システム運用の健全性に関するメトリックをリアルタイムで提供するため、TotalAgility プラットフォームの管理と評価の改善に役立ちます。

キャプチャ クライアント

大規模なジョブを処理しながら UI の最適な応答性とパフォーマンスを実現するには、次の項目の使用をお勧めします。

1. **Chrome** ブラウザ: Kofax TotalAgility はサポートされているすべてのブラウザで可能な限り最高のパフォーマンスを達成しようとしますが、Kofax TotalAgility は Chrome ブラウザで最高の UI 応答性を実現します。これは特に、大きなジョブで作業しているときや、ドック解除されたイメージビューア (またはドック解除された Web Capture コントロール) を使用しているときに顕著になります。
2. リモート スキャン: リモート サイトでのスキャン スループットを向上させるには、Scan Agent サービスを展開して、非同期ジョブ アップロードを有効にします。

オブジェクト ID またはオブジェクト インデックスの収集をサポートする API

一般的に、SDK API 呼び出しではデータベースの往復を行います。複数のオブジェクト (フォルダ、ドキュメント、またはフィールドなど) へのアクセスが必要な場合は、オブジェクト ID またはそのインデックスの収集を渡すことができる API のバージョンを使用することをお勧めします。データベース通信が大幅に少ないため、複数バージョンの API は、同じ API の単一バージョンの複数の呼び出しよりも効率的に実行されます。

以下は、複数バージョンの API の例です。

ドキュメント メソッド

- MergeDocuments
- RejectDocuments
- UnRejectDocuments
- UpdateDocumentsProcessingCompleted
- DeleteDocuments

ページおよびフィールド関連のメソッド

- RejectPages
- UpdatePages
- UpdateWords
- ValidateDocumentFields
- ValidateFolderFields

第3章

プッシュ アクティビティ処理時の Transformation Server インスタンスの構成

プッシュ アクティビティは、数多くの自動ステップを実行するように設計されたプロセスで、同じジョブ内で実行する次のアクティビティをユーザーに返すときに生じるものです。Kofax TotalAgility では、Transformation Server が作業のためにポーリングする通常の処理の代わりに、分類、抽出、イメージ処理などのアクティビティは Transformation Server にパージされます。

処理を許可されている任意の Transformation Server インスタンスでプッシュ アクティビティを処理できます。

- プッシュ アクティビティを処理する Transformation Server を制限するには、プッシュ アクティビティを処理する必要のないサーバー上で「EnableSynchronousCalls」を false に設定します。
- 通常のアクティビティではなく、プッシュ アクティビティのみを処理する必要があるサーバーの場合は、「SynchronousOnlyProcessing」を true に設定します。
- 通常のアクティビティとプッシュ アクティビティの両方を処理する必要があるサーバーでは、「ReservedSlotsNumber」を 1 または 2 に設定して、時折発生するプッシュ アクティビティがキューに留まらないようにします。

第 4 章

セキュリティ

セキュア変数の使用、権限の割り当て、ウイルス スキャナの使用、Microsoft のセキュリティ ベスト プラクティスのインストール前での適用などの対策を講じて、セキュリティを強化します。

保存データの保護

セキュア サーバー変数を使用して、サーバー変数内に保存されている機密情報を保護します。

SQL Server TDE を使用してデータベースを暗号化し、セキュリティを強化します。データベースの暗号化はパフォーマンスに影響します。したがって、主要な TotalAgility データベースやドキュメント データベースなど、機密情報を含むデータベースのみを暗号化することを検討する必要があります。

TotalAgility の構成ファイルには、機密情報などの接続文字列への参照が含まれています。.NET ユーティリティを使用して構成ファイルを暗号化することにより、これらの詳細を保護できます。『Kofax TotalAgility インストール ガイド』の「TotalAgility 構成ファイルの暗号化と復号化」セクションを参照してください。

送信中のデータの保護

HTTPS 経由の送信データ

TotalAgility フォーム内で使用されるグローバル変数は、ブラウザで表示および編集できます。TotalAgility フォームでセキュア グローバル変数を使用して、送信中の機密データを保護します。HTTPS を使用することで、特に公開 Web サイトでの展開でのセキュリティを強化できます。

注 HTTPS を使用するには、有効な証明書が必要です。

公開 Web サイトが必要な場合は、ファイアウォールで保護された非武装地帯 (DMZ) に Web サーバーを展開します。Web サーバーとアプリケーション サーバー間のトラフィックを保護する必要がある場合は、アプリケーション層で HTTPS を使用することもお勧めします。セキュリティをさらに強化するには、Web 層とアプリケーション層の間に追加のファイアウォールを配置します。

データベースで転送するデータ

HTTPS 経由の送信データを保護することは最も重要ですが、データベースで送受信するデータは、SQL Server の接続暗号化を使用して保護をさらに強化できます。

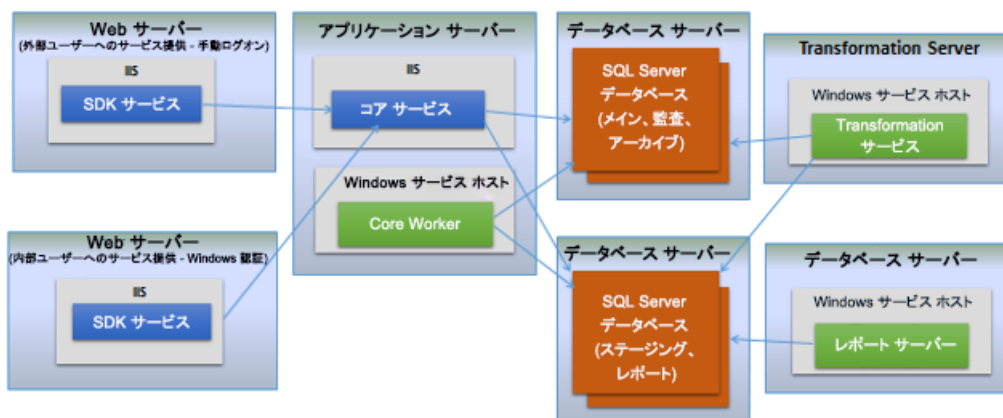
権限

セキュリティを強化するには、「最小限の権限の原則」を適用します。

内部ユーザーと外部ユーザーへのサービス提供

内部ユーザー (LAN 内) と外部ユーザー (インターネットまたは公開 Web サイト) にサービスを提供するには、Web サーバー上で異なる Web サイトを構成する必要があります。1 つは内部ユーザー用、もう 1 つは外部ユーザー用です。たとえば、イントラネット上の Web サイトには、通常は、公開 Web サイトと同じセキュリティ制限や要件はありません。

セキュリティ要件ごとに異なる Web サーバーを使用します。



Scan Cient での悪意のあるファイルのアップロードに対する保護

TotalAgility は、Web ブラウザを介したファイルのアップロードを処理するために Web アーキテクチャを使用しているため、Web サーバーのセキュリティ保護と同様の方法で、TotalAgility を悪意のあるファイルのアップロードから保護する必要があります。OWASP (Open Web Application Security Project) の推奨事項に基づいて、次のベスト プラクティスをお勧めします。

1. インストール前に、Microsoft による次のセキュリティ ベスト プラクティス (Microsoft IIS 7 の Web サイトを参照) を実装してください。
 - Web サーバーのセキュリティを構成する
 - セキュリティの構成
 - IIS 操作ガイド
 - 組み込みユーザーおよびグループのアカウントを理解する
 - IIS セキュリティ チェックリスト

2. 許容ファイル サイズを制限します。Web サーバー自体を介してファイル サイズの制限を構成します。制限を設定する前に、受信するファイルのサイズを明確に理解してください。
3. サーバーでウイルス スキャナを使用し、ファイアウォール セキュリティ アプライアンスまたはウイルス スキャンをサポートする Web アプリケーション ファイアウォールを展開します。例:
<https://techlib.barracuda.com/waf/antivirusprotection>

サーバーの強化

セットアップに基づいて、一般的なサーバー攻撃から保護するために、次の対策の一部またはすべてを適用できます。

クロスサイト スクリプティング保護の提供: このヘッダーは、ほとんどの最新のブラウザで用いられている組み込み XSS 保護を構成します。このヘッダーにより、ブラウザは攻撃を検知すると応答をブロックします。

IIS で、次のように HTTP 応答ヘッダーを TotalAgility Web アプリ レベルに追加します。

```
X-Xss-Protection=1; mode=block
```

URL スキャン モジュールのインストール: このセキュリティ ツールは、IIS が処理する HTTP 要求のタイプを制限します。特定の HTTP 要求をブロックすることにより、UrlScan セキュリティ ツールは、潜在的に有害な要求がサーバー上のアプリケーションに到達するのを防ぎます。管理者が設定したルールに基づいて要求をフィルタリングすることにより、サーバーに送られるすべての要求をスクリーニングします。要求のフィルタリングにより、有効な要求のみが処理されるため、サーバーのセキュリティ保護に役立ちます。HTTP クエリ文字列値およびその他の HTTP ヘッダーをフィルタリングして SQL インジェクション攻撃を軽減させながら、アプリケーションで根本原因を解決するという構成が可能です。

MIME タイプのセキュリティ リスクの軽減: 一部のタイプのドライブバイダウンロードに対してサイトのセキュリティを改善するには、次のヘッダーを IIS の TotalAgility Web アプリ レベルでサイトに追加します。

```
X-Content-Type-Options=nosniff
```

IIS サーバー レベルでクライアント キャッシュを無効化: 管理者としてコマンド プロンプトを開き、次のコマンドを実行します。

```
appcmd set config /section:staticContent /clientCache.cacheControlMode:DisableCache
```

IIS レベルで、キャッシュ関連の HTTP 応答ヘッダーを追加します。

```
Cache-Control=no-cache; no-store; must-revalidate  
Pragma=no-cache  
Expires=0
```

HTTP Strict Transport Security (HSTS) を構成: このメカニズムは、プロトコル ダウングレード攻撃や Cookie ハイジャックから Web サイトを保護するのに役立ちます。プロトコル ダウングレード攻撃は、https などの暗号化接続をサーバーに放棄させ、安全でない接続 (http) を優先させようとしています。これにより、Cookie ハイジャックが容易になり、攻撃者は有効なセッション キーにアクセスできるようになります。HSTS を実装するには、サイトに次のように web.config を構成します。

```
<?xml version="1.0" encoding="UTF-8"?>  
<configuration>  
<system.webServer>  
<rewrite>  
<rules>
```



```

<rule name="HTTP to HTTPS redirect" stopProcessing="true">
  <match url="(.*)" />
  <conditions>
    <add input="{HTTPS}" pattern="off" ignoreCase="true" />
  </conditions>
  <action type="Redirect" url="https://{HTTP_HOST}/{R:1}" redirectType="Permanent" />
</rule>
</rules>
<outboundRules>
  <rule name="Add Strict-Transport-Security when HTTPS" enabled="true">
    <match serverVariable="RESPONSE_Strict_Transport_Security" pattern=".*" />
    <conditions>
      <add input="{HTTPS}" pattern="on" ignoreCase="true" />
    </conditions>
    <action type="Rewrite" value="max-age=31536000" />
  </rule>
</outboundRules>
</rewrite>
</system.webServer>
</configuration>

```

コンテンツセキュリティポリシーヘッダーの追加: このヘッダーにより、Web サイトのコンテンツの承認されたソースのホワイトリストを定義できます。ブラウザがサイトにロードできるアセットが制限され、クロスサイト スクリプティング攻撃への対策として機能します。IIS レベルで、次の HTTP 応答ヘッダーを設定します。

```
Content-Security-Policy=default-src https: data: 'unsafe-inline' 'unsafe-eval'
```

クリックジャッキング対策の提供: サイト上の iframe の動作を制御して、クリックジャック攻撃を防ぐことができます。自分のサイトからの iframe のみを許可するように X-Frame Options ヘッダーを設定します。

```
X-Frame-Options=SAMEORIGIN
```

OPTIONS 動詞の無効化: OPTIONS 動詞は、役立つ情報を攻撃者に提供するため、さらなる攻撃を促進してしまいます。Web サーバーでは、組み込み IIS モジュールを使用します。

```
Request Filtering -> HTTP Verbs -> Deny Verbs...
```

不要なヘッダーの削除: 攻撃者は、偵察段階で X-Powered などの非機能ヘッダーを使用して、ターゲットとなるプラットフォームを識別することができます。IIS HTTP 応答ヘッダー モジュールを使用して、このようなヘッダーを削除します。

悪意のあるデータベース接続と Web サービス URL に対する保護

TotalAgility ソリューションで外部データベース接続または Web サービス URL を使用する場合、悪意のあるデータベース接続や Web サービス URL に対して TotalAgility を保護する必要があります。TotalAgility Designer のシステム設定のホワイトリスト構成で、[すべて許可] オプションをクリアし、TotalAgility ソリューションで使用する必要のある各データベース接続文字列と Web サービス URL を指定します。

第 5 章

データベース

この章では、データベースに関連するベスト プラクティスについて説明します。

サイジング

要件に合ったデータベース サーバーを採用します。データベースのサイズは、ジョブやドキュメントなどのデータの全体的なスループットとソリューションの実装に左右されます。たとえば、多数の長期にわたるケースを含むソリューションでは、より大きなデータベースが必要になる場合があります。

適切に TotalAgility 保持ポリシーを採用し、データベースの拡大を管理します。

監査が不要な場合は、データベース サイズが不必要に大きくなるようにスイッチをオフにします。

展開

- 仮想サーバーではなく、専用の物理的なマシンにデータベース サーバーをインストールします。
- コア Kofax TotalAgility データベースとは別のデータベース サーバーにレポート データベースをインストールします。これにより、最適なパフォーマンスを確保しながら、コアの TotalAgility データベース サーバーの負荷を軽減します。
- ディスク I/O の競合を制限するために、ドキュメント データベースを独自の物理的なディスク上に置く必要があります。
- データベースとトランザクション ログを異なるドライブに配置します。物理的にドライブを分けることが好まれます。
- MSSQL ファイルを合理的な複数のドライブに配布します。サンプル構成を参照してください。
 - SQL Server のインストール – E:
 - SQL Server データベース データ ファイル – D:
 - SQL Server データベース インデックス ファイル – I:
 - SQL Server トランザクション ログ – L:
 - SQL Server TempDB データ ファイル – T:
 - バックアップ – F:

注 上記の各項目のボリューム サイズと 1 秒あたりの入出力操作 (IOPS) 要件は、システム上のドキュメントとユーザー ボリュームによって異なります。

- MSSQL ファイルを含むすべてのディスクが 64K ブロック サイズでフォーマットされていることを確認してください。

- データベース ファイルのすべてのファイルの増加が、デフォルトの割合ではなく設定値であることを確認します。推奨: データ、インデックス、およびトランザクション ログ ファイル用に 256 MB。
- すべての SQL Server サービスのサービス アカウントを構成します。MSSQL インストーラーは、インストール中にサービス アカウントに最小限の特権を自動的に割り当てます。
- SQL Server エンジンのみをインストールします。アナリティクス サービス、レポート サービス、統合サービスなどの追加コンポーネントは必要ありません。
- MSSQL をインストールするときは、組織で使用されている必要なすべてのサービスパックと累積的な更新プログラムを必ず適用してください。

注 MSSQL の非本番インスタンスは、本番と同じ更新プログラムである必要があります。

- 必ず、TCP/IP プロトコルを有効にしてください。
- TempDB を構成します。
 - デフォルトでは、MSSQL は、MSSQL がインストールされているドライブに TempDB を配置します。TempDB をその適切な場所に移動します。
 - 競合を避けるために、TempDB に複数のデータ ファイルを追加することを検討してください。
 - TempDB を最終サイズに手動で拡張します (サイズは実装によって異なります)。
 - TempDB ファイルの自動拡張を有効にしないでください。
- 混合モード認証が有効になっていることを確認します。

メモリ

OS および SQL Server のその他のインスタンスに必要なメモリ (およびコンピュータが SQL Server 専用ではない場合は他のシステムの使用量) を物理的なメモリの合計から差し引いて、SQL Server に割り当てることができるメモリの最大量を判断します。SQL Server のメモリ使用量の詳細については、Microsoft の Web サイトを参照してください。

メンテナンス

定期的なデータベース保守計画をスケジュールして、インデックスの断片化、統計、バックアップ、アーカイブを管理します。通常、15 分間隔のトランザクション ログ バックアップは適切な頻度で発生します (ログが制御不能にならないようにします) が、パフォーマンスに影響を与えるほど頻繁ではありません。

MSSQL サーバーでウイルス対策プログラムを実行できますが、ウイルス スキャンによるパフォーマンスのボトルネックを防ぐために特定のファイルを除外することができます。除外の完全なリストについては、Microsoft の Web サイトを参照してください。

回復機能

SQL Server のクラスタリング、レプリケーション、またはミラーリングを採用し、データベースの回復機能を検討します。高可用性セクションを参照してください。

MSDTC

トランザクションが複数のデータベースにまたがる場合、MSDTC を有効にします。MSDTC は、メインデータベース テーブルとアーカイブ データベース テーブルが異なるデータベースに分けられているときに、TotalAgility に限り必要です。詳細については、『Kofax TotalAgility インストール ガイド』の「データベースのインストール」セクションを参照してください。

第 6 章

ソリューションの構築

この章では、TotalAgility でソリューションを構築するためのベスト プラクティスについて説明します。

ソリューションを構築するときは、後の段階での UX の再作業を避けるために、まず、ドキュメント、プロセス、ルールなどの非ユーザー インターフェイス面に重点を置きます。TotalAgility は、ほとんどの場合、フォームなしでデータを表示するため、多くの UX を構築することなく、フローと操作を早期に再生してテストできます。これにより、早期のシステム テストも容易になります。

システム設定

TotalAgility は、多くのシステム設定がデフォルトのままインストールされます。確認し、必要に応じて変更し、本番環境に合わせる必要があります。

これらの設定の説明と構成については、『Kofax TotalAgility のヘルプ』の「システム設定」を参照してください。

システム セッション ID

TotalAgility のインストールごとに、一意のシステム セッション ID が生成されます。システム セッション ID を使用することで、セキュアな SDK 呼び出しを実行できます。

セキュリティ侵害があった場合 (例: 誰かが許可なくセッション ID を使用してシステムにアクセスした場合) にセッション ID を再生成したり、セッション ID の値を任意に設定 (例: 開発環境から本番環境に移行する際に、両方の環境で同じセッション ID を使用) したりできます。

注 値を更新する場合は、Web.config ファイルを更新する必要があります。詳細については、『TotalAgility のヘルプ』の「システム > システム設定 > ログオンおよび認証 > ユーザー セッション」セクションを参照してください。

セッション タイムアウトおよびバッチ セッション タイムアウト

タイムアウト設定には、セッション タイムアウトとバッチ セッション タイムアウトが含まれます。

セッション タイムアウト

セッション タイムアウトのデフォルト設定は 1 時間です。この期間の後 (ユーザーの最後のアクティブ日に関連)、システムは TotalAgility のユーザー セッションを自動的に無効にします。

TotalAgility をタイムアウトさせるには、適切な値を設定するか、システム タスク「セッション タイムアウトの処理」を無効にします。

セッション タイムアウトを使用する場合、セッションには有効期限があり、一定期間が経過すると期限切れになります。セッションが頻繁に期限切れにならずにユーザーが操作を快適に完了することができるように、アプリケーションの目的および性質に合わせて、セキュリティと使いやすさが両立するように適切に値を設定することをお勧めします。

ユーザーが複数のセッションを持つことを許可または制限することができます。「[複数のユーザー ログオンを許可](#)」を参照してください。

バッチ セッションのタイムアウト

タイムアウト時間のデフォルト設定は 30 分です。タイムアウト設定は、スキャン、Validation、Verification、Document review などのキャプチャ関連タスクを実行するときに特に使用されます。

キャプチャ バッチをタイムアウトにする場合は、適切な値を設定するか、「キャプチャ タイムアウトの処理」システム タスクを無効にします。

一般的なユーザーがキャプチャ関連のタスクを完了するのにかかる時間およびキャプチャ作業の負荷を考慮して、セッション タイムアウトを設定することをお勧めします。たとえば、一般的なユーザーが一度にスキャンまたは検証する必要があるドキュメント数を考慮します。

次の点に注意します。

- バッチは、セッション タイムアウトとは無関係にタイムアウトできるようにして、セッション タイムアウトよりも短い間隔にする必要があります。
- ステイッキー セッションを使用する場合、バッチ セッション タイムアウト間隔は適用されません。代わりに、ASP.NET セッション タイムアウト間隔が使用されます。したがって、ステイッキー セッションを使用する場合は、システムタスク「キャプチャ タイムアウトの処理」をオフにすることをお勧めします。ステイッキー セッションの詳細については、『Kofax TotalAgility 管理者ガイド』を参照してください。

パスワード形式

このプロパティは、ユーザー パスワードの長さや複雑さを制御できる正規表現です。

デフォルトのパスワードの長さは 10 文字です。

複数の文字セットの文字を組み合わせた 8 文字以上のパスワードに限り受理するようにすることをお勧めします。

パスワード ハッシュ アルゴリズム

事前構成済みの Scrypt 暗号化アルゴリズムの使用を強くお勧めします。

SHA-1 も利用可能ですが、暗号解析の最近の進歩により、SHA-1 アルゴリズムには弱点が見つかっています。Scrypt の方がはるかに強力です。

パスワードを使用しないログオンの無効化

TotalAgility SDK (およびデプロイ済み TotalAgility エンドポイント) について知識のある人は、対象ユーザーのユーザー名のみを知っていると有効なユーザーに代わって、TotalAgility セッションを取得できます。

すべての非認証「セッション取得」 SDK メソッドを保護するには、「パスワードを使用しないログオンの無効化」設定を有効にします。

この設定が有効な場合:

- セッション ID を取得する API は、有効なパスワードなしで呼び出すことができません。
- パスワードなしの既存のログオン呼び出しはすべて失敗します。

したがって、ソリューション コンテキストを検討し、影響があるかどうかを判断し、それに応じて設定を有効にする必要があります。

パスワードのリセット

パスワードによる手動認証を使用している場合は、パスワードリセットの要求を処理するシステム プロセスを設定します。

ユーザーが有効な電子メール アドレスを持っていることと、SMTP サーバーが構成されていることを確認してください。

複数のユーザー ログオンを許可

ユーザーが同時に複数の TotalAgility セッションにログオンできるようにするには、[複数のユーザー ログオンを許可] 設定を選択します。詳細については、『TotalAgility のヘルプ』の「システム > システム設定 > ログオンおよび認証 > パスワードとログオン」セクションを参照してください。

デフォルトでは、TotalAgility は、ユーザーごとにアクティブなセッションを 1 つだけ許可します。ユーザーが別のブラウザや別の場所からログオンすると、最初のセッションは終了します。

複数セッションのサポートを有効にすると、各ユーザーが個別に動作する複数のセッションを利用することができます。たとえば、それぞれのセッションで、他のセッションに影響を与えずにタイムアウトやログアウトを行えます。

ただし、追加のセッションごとに同時ユーザー ライセンスを利用します。そのため、一部のシナリオでは、複数のログオンを許可して使いやすさを向上させることが理にかなっている場合がありますが、経験則から言えば、必要以上の機能を許可しないことです。

ユーザーが同時に複数セッションに接続することがない場合、複数のログオンを許可しないように設定し、許可されていないユーザーからの攻撃のリスクを軽減します。

特にログアウトせずにブラウザを閉じるとセッションが開いたままになり、セッションがタイムアウトするまで同時ライセンスを利用するため、ライセンスが原因で複数のログオンを許可しないことは理にかなっています。

アカウント ロックアウト ポリシー

TotalAgility 内で、誤ったパスワードを使用して手動でログインした場合のログオン試行の失敗に対するシステムの処理方法を構成できます。

システムのデフォルトを使用することをお勧めします。

ログオン試行の最大回数: この設定では、ユーザー アカウントがロックアウトされるまでのログオン試行の失敗回数を決定します。しきい値セットは、運用効率とセキュリティのバランスであり、組織のリス

ク レベルに依存します。ユーザー エラーを許可し、悪意のある攻撃を防ぐには、組織の許容可能な開始点として、4 よりも上で 10 未満の設定 (デフォルト値は 5) を維持します。

アカウント ロックアウト期間: この設定では、アカウントが自動的にロック解除されるまでロックアウトされたままである分単位の数 (デフォルト値は 30 分) を決定します。

アカウントが自動的にロック解除されないように、この値を 0 に設定できます。それは良いアイデアのように思えるかもしれませんが、ただし、そのようにすると、誤ってロックされたアカウントのロックを解除するために管理者へのリクエスト数が増える可能性があります。

ビジネス カレンダー

TotalAgility Workspace には、すべてのリソースの稼働日、非稼働日、および稼働時間を設定するために使用できる、グローバルなビジネス カレンダーがあります。また、各リソースには、グローバル カレンダーから抽出した個人用のカレンダーがあります。

デフォルトでは、ビジネス カレンダーはオフになっています。ビジネス カレンダーがオンになっていると、ジョブの期間やアクティビティの期限などの日付の計算が、労働時間に関連して実行されます。

アプリケーションで日付を考慮する必要がある場合、または満たすべきサービス レベル契約がある場合は、カレンダー機能を有効にすることをお勧めします。

電子メール アドレスの重複を許可する

TotalAgility では、複数のリソースによる同じ電子メール アドレスの使用を許可または禁止できます。

複数のリソースの同じ電子メール アドレスの使用は、テスト目的の開発および UAT 環境に適切です。さらに、本番環境では、複数のリソースが個々の電子メール アドレスではなく、同じ配布電子メール アドレスを使用する可能性があります。同様に、複数のグループが同じ配布電子メール アドレスを使用する場合があります。

ただし、自分の電子メール アドレスを使用した外部ユーザー ログインがある場合は、重複を許可しないことをお勧めします。

監査ログへの書き込み

監査はデフォルトでオンになっています。監査ログはサーバーのアクティビティとパフォーマンスの監視に役立ちますが、監査データにより TotalAgility データベースが大幅に増大します。

「監査ログに書き込み」設定をオフにして、データベース サイズを減らしてパフォーマンスを向上させることができます。

ただし、監査ログを保持する場合は、関連する保持ポリシーを構成して、古い監査ログ エントリを削除します。

終了済みジョブのアーカイブ

終了したジョブを別のテーブルに移動し、別のデータベースに保持して、データベース サイズを管理します。

「終了済みジョブのアーカイブ」設定がデフォルトで選択されています。結果として、ジョブが完了すると、システム タスクである「ジョブのアーカイブ」でアーカイブ テーブルにジョブを移動します。

これにより、ライブ ジョブ テーブルが最小限になり、継続的な増大を抑えることができます。

終了したジョブをアーカイブ テーブルに移動するか、ジョブの終了後に (または定期的に) パージすることをお勧めします。

注 一部のジョブの履歴が不要な場合は、プロセスの「履歴の記録」設定をクリアします ([プロセスのプロパティ]> [履歴、レポート、および実行] タブ)。

レポート

プロセス データに対して分析を実行し、Insight でデータが取得されるようにする場合は、プロセスごとに「分析に含める」設定を選択します ([プロセスのプロパティ]> [履歴、レポート、および実行])。

キャプチャ データはレポート データベースに保存されます。ETL エージェントがシステム タスク (抽出/変換/ロード) を処理してステー징からウェアハウスにデータを変換する頻度を制御するには、[システム]> [システム設定]> [データベース、保持、およびレポート]> [レポート サーバー]> [レポート ウェアハウス ETL エージェント] にあるシステム設定を使用して間隔を設定できます。

この変換を夜間に限定することで、日々のパフォーマンスに影響しないようにすることができます。

フォームのキャッシュ

フォームのキャッシュはデスクトップ フォームにのみ適用されます。デフォルトでは、フォームのキャッシュはオンになっています。

ユーザーが同じフォームを繰り返し使用する場合は、フォームのキャッシュを有効にすることをお勧めします。フォームをキャッシュすることで、クライアント側のブラウザでの読み込み速度が速くなり、読み込みパフォーマンスが向上します。

キャッシュに追加される最近使用されたフォームの数を指定します。たとえば、10 は 10 個のフォームがキャッシュされることを意味します。

デフォルトのフォーム キャッシュ サイズは次のとおりです。

- フォーム: 20
- ドキュメント: 6
- フォルダ: 10

一般的な条件下では、デフォルトの設定にすることで、パフォーマンス要件とクライアントのストレージ要件のバランスが最適になります。

フォームは、表示されたときにクライアントのキャッシュに追加されます。キャッシュがフォーム タイプでいっぱいになると、アクセス日時の最も古いフォームがキャッシュから削除され、新しいフォーム用のスペースが確保されます。

フォームがキャッシュからロードされるたびに、キャッシュから削除されないよう、そのフォームのキャッシュ内の最終アクセスが更新されます。

これは、最も頻繁に表示されるフォームがキャッシュにあることを意味します。

リフレッシュ間隔

カスタム ページとイメージを TotalAgility にアップロードして、フォームで使用できます。実行時に表示されるイメージまたはカスタム ページは、データベースからダウンロードされ、キャッシュされます。

イメージまたはカスタム ページが TotalAgility Designer で更新されると、キャッシュがリフレッシュされるまで最新バージョンはユーザーに表示されません。

リフレッシュの頻度を制御するには、次の 2 つの設定「イメージの更新時間」と「カスタム ページの更新期間」を使用できます。どちらもデフォルトの設定は 120 分です。

イメージまたはページが更新される可能性を考慮して、間隔を設定することをお勧めします。運用環境では、更新される可能性が低い場合、間隔の値をより大きく設定できます。

スキル レベル

デフォルトでは、サーバー ベースのスキル レベルが選択されています ([システム] > [システム設定] > [プロセス] > [SLA および作業の割り当て])。つまり、ジョブ内のタスクごとにリソースのスキル レベルがチェックされます。この場合、リソースはすべてのプロセスに適用される 1 つの値を持ちます。リソースは、適切なスキルを持つアクティビティでのみ作業できます。

プロセス ベースのスキルを使用する場合、プロセスごとに、リソースに特定のスキルが割り当てられます。これには、より多くのメンテナンスが必要になる場合があり、データベースの負荷が大きくなります。

スキル レベルを使用しない場合、データベース アクセスの複雑さが軽減されるため、パフォーマンスが向上します。

例外処理

ソリューションに活用できるシステム レベルの例外があるかどうかを検討します。たとえば、Web サービスの呼び出しが失敗した場合、ジョブの一時停止に加えて、他の処理を実行したいことがあります。

ソースに関係なく例外を処理するようにシステム レベルで例外処理を構成することも、プロセスに固有の例外を構成することもできます。

デフォルトでは、システムによって例外が処理されません。そのため、ジョブの処理中にエラーが発生すると、ジョブは一時停止し、通知は送信されません。

ダウンタイムを厳密に監視して回避する必要がある高価値のユースケースの場合、またはタイムクリティカルな作業を実行する必要がある場合は、アプリケーションの目的と性質に合わせて例外を適切に構成することをお勧めします。

例外の発生のお知らせだけ行うなど、例外処理プロセスを非常に一般的なものに構成することも、マップに渡される初期化データを使用するようにカスタマイズすることもできます。この方法は、すべての一時停止に対して通知を送信しない場合、または特定のタイプの例外に対して修復アクションをカスタマイズする場合に役立ちます。

保持ポリシー

優れた設計とメンテナンスにおいては、不要なデータベース増大を防ぐために、古い項目や使用されていない項目をシステムから手動で削除することを推奨する必要があります。

保持ポリシーにより、データベース サイズの最小化を自動的に行うことができます。デフォルトではシステムに保持ポリシーが設定されていないため、すべてのアーティファクトは手動で削除しない限りデータベースに無期限で保持されます。

使用されていない項目のクリーン アップ、データベースの増大の管理、パフォーマンスの最適化のために、保持期間を定義することをお勧めします。

チームベースの展開の構成管理

チームベースの環境でソリューションを開発するには、構成管理を使用する必要があります。Kofax TotalAgility と構成管理ツール間に直接的な統合がない場合、次のアプローチを利用できます。

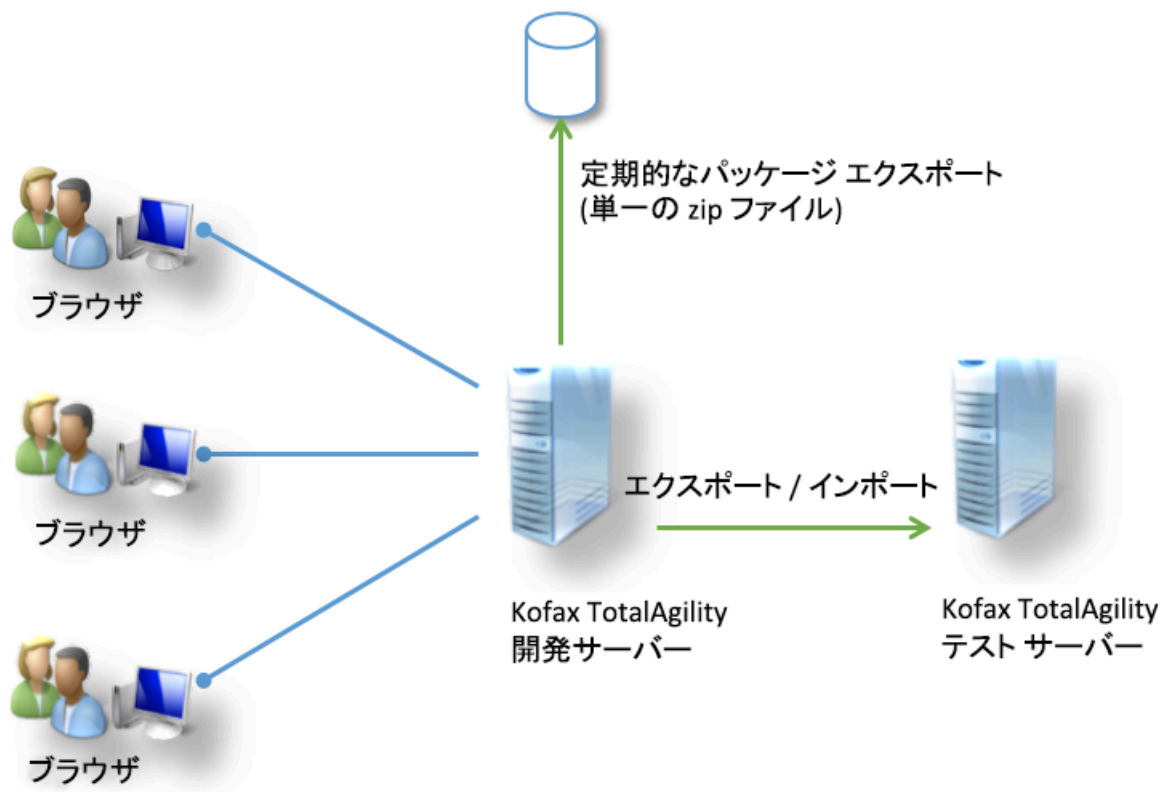
- 共有開発環境: セントラル サーバー上に一度インストールされた Kofax TotalAgility
- スタンドアロン開発環境: ローカルに複数回インストールされた Kofax TotalAgility

このアプローチではチェックインの衝突によるダウンタイムが最小で最も生産性が高いことがわかっているため、スタンドアロン アプローチを使用することをお勧めします。

組織に最適なアプローチを決定するには、以下のセクションで説明されている分析を確認します。

共有展開環境

開発チームは、開発サーバーの TotalAgility Designer を参照し、設定なしで使えるロック機能を使用して、フォーム、プロセス、ビジネス ルール、およびその他の関連するアーティファクトへのアクセスを制御します。このアプローチでは、他の開発者が作業を継続できなくなる競合する変更によって生じるシステムのダウン タイムを最小限に抑えるために、開発チームにはかなりの規律が必要です。



長所

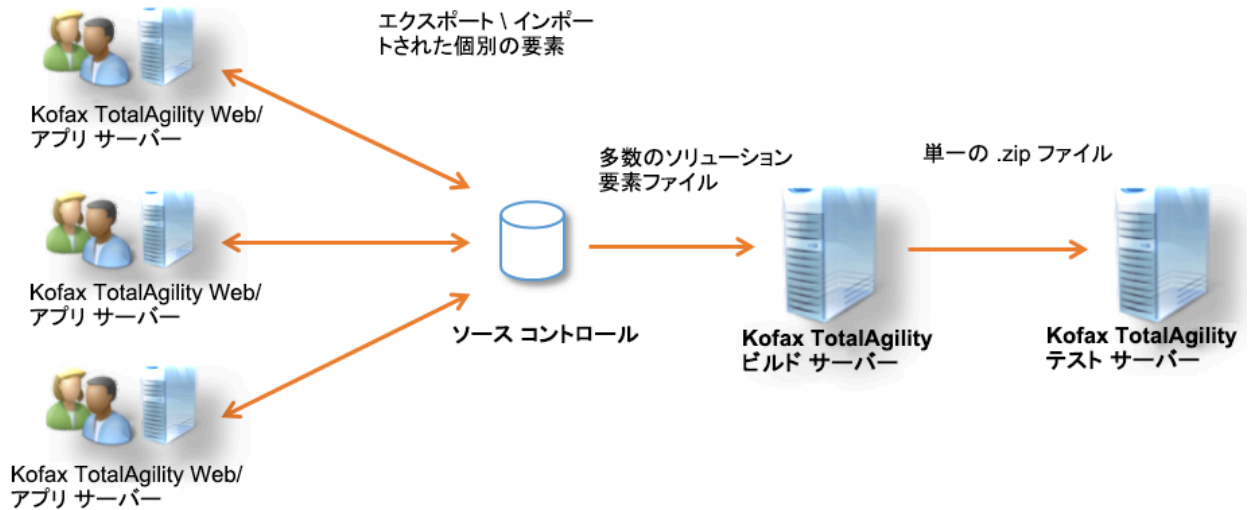
- 維持する開発環境が少ない。

短所

- すべてのアクティビティはサーバーで発生するため、機能を妨げる変更は、他の開発者に影響を与え、遅延の原因になる可能性があります。
- サーバーで実行できるカスタム .NET コードのバージョンは 1 つのみのため、他の開発者に影響を与えずにテストすることは困難です。
- すべての開発者は、単一の開発サーバーの可用性に関連付けられています。

スタンドアロン展開環境

開発チームが Kofax TotalAgility をインストールする開発マシンの数に制限はないため、開発者は、実装およびテストする変更非常に柔軟に対応できます。



ソリューションに加えられた変更を管理および追跡するには、次のプロセスを適用します。

1. 製品のエクスポート機能を使用して、プロセスやフォームなどの項目ごとに個別の ZIP ファイルを作成します。
2. Team Foundation Server などの構成管理ツールに ZIP ファイルを配置します。
3. チーム メンバーをビルド マスターとして任命して、以下を管理します。
 - ZIP ファイルの更新
 - グローバル変数の追加または更新
 - テーマと CSS の追加または更新
 - ナビゲーション メニューなどの追加または更新
4. 開発者は、変更された要素を含むエクスポートされた ZIP ファイルの形式でファイルと要素をビルド マスターに送信します。ビルド マスターが理解しやすいように変更内容のメモを ZIP ファイルに含めます。
5. 開発者は毎日、構成管理ファイルから ZIP ファイルをチェックアウトし、必要な変更を実装し、完了したらチェックインします。
6. 毎日、ビルド マスターは少なくとも次のことを行う必要があります。
 - 構成管理から最新の ZIP ファイルをインポートします。
 - 新しい項目でパッケージを更新し、パッケージをエクスポートします。
 - 開発者と品質アナリストに、更新された作業パッケージが利用可能であることを知らせます。

新しいリリースを新しい環境にプロモートする

ソリューションを開発サーバーからテストサーバーまたは本番サーバーに移行する場合は、パッケージ機能を使用します。パッケージを使用すると、ソリューションに関連付けられているすべての項目をすばやくエクスポートできます。このプロセスを簡単にするために、次の機能を使用できます。これにより、項目の見落としがなくなります。

- プロセスに関連付けられた項目を自動的に含める。
- フォームに関連付けられた項目を自動的に含める。

- カテゴリに関連付けられた項目を自動的に含める。
- パッケージを含める (ソリューションの後続のリリースの一部として変更された項目に対して、個別のパッケージを作成される場合があります)。
- パッケージをシステム上の項目と比較する。
- 最近変更された項目を検索する。
- ターゲット サーバーへのインポート: 接続文字列など、グローバル変数とサーバー変数の値を、ターゲット環境の適切な値に更新します。

各パッケージにアクセス許可を割り当てて、他のユーザーによる不必要な変更を防ぎ、問題が生じないようにします。

リソース

アクセス許可を許可または拒否し、要件に基づいて作業割り当てを構成します。

アクセス許可

アクセス許可を管理して、Designer とプロセスへのリソースのアクセスを許可したり制限したりします。

Designer

システム設定から TotalAgility Designer 内の機能の主要領域へのアクセスを制限し、プロセス、フォーム、またはより重要なシステム設定を変更できるユーザーに対して制限を強化します。

「全員」にアクセスを許可すると、誤った変更が適用され、システムが予期せぬ動作をする可能性があります。デフォルトでは、管理者グループのメンバーのみがこれらの領域にアクセスできます。たとえば、Designer に適切な権限が付与されていないリソースは、Designer を開いてプロセス、フォーム、およびその他の領域を表示することができません。

Designer のアクセス許可構成に関する詳細については、『Kofax TotalAgility のヘルプ』のトピック「TotalAgility のさまざまな領域に対するアクセス許可の割り当て」を参照してください。

プロセス

プロセス内のメンテナンス アクセス機能を使用して、機密性の高いプロセスへのアクセスを制限します。これにより、プロセスへの変更をコントロールし、徹底的にテストされ承認された後のみ使用できるようにします。単一のリソースだけがメンテナンス アクセス権を持つことができるため、この目的のためにグループを構成することをお勧めします。

プロセスのアクセス許可構成に関する詳細については、『Kofax TotalAgility のヘルプ』の「メンテナンス アクセス」を参照してください。

アクティビティの割り当て

作業は、保留状態になるとすぐに自動的に割り当てられるか、スーパーバイザーにより手動で割り当てられるか、特定の時間に割り当てられるようにスケジュールされます。手動の介入を必要としないため、自動作業割り当てをお勧めします。

手動での作業割り当て

作業する前にスーパーバイザーがリソースに作業を明確に割り当てる必要がある場合は、アクティビティの [割り当て] プロパティを選択します。スーパーバイザーが手動で特定のリソースに作業を割り当てるまで、アクティビティは保留になりません。手動での割り当てに関する詳細については、『TotalAgility Workspace のヘルプ』の「作業の割り当て」トピックを参照してください。

スケジュールされた作業割り当て

作業を割り当てるプロセスを自動化するには、システム カテゴリで利用可能な「システムの自動作業割り当て実行」プロセス マップを使用します。このマップは、適切な間隔でジョブ スケジューラを使用して実行するようにスケジュールでき、必要に応じて変更できます。

自動アクティビティ割り当て

適切な作業が保留状態になるとすぐに適切なリソースに割り当てられ、ユーザーの生産性が向上するため、自動アクティビティ割り当てをお勧めします。

次のシナリオでは、使用する最も適切なリソース割り当て機能を強調しています。

静的リソースへの割り当てと動的リソースへの割り当て

どのリソース グループまたは個人がアクティビティを実行できるかを事前に知っている場合は、静的割り当てを使用します。それ以外の場合は、動的割り当てを使用します。たとえば、アクティビティが従業員の雇用に関連している場合、HR リソース グループがそのアクティビティを実行するように、静的割り当てが推奨されます。

リソースを構成するときは、グループまたはロール (固定または変動) を使用して、実行時の割り当てをさらにコントロールします。

ルールがあるとさらに柔軟性が高まるため、リソース割り当てがより複雑な場合は、ルールを適用します。ルールを使用したリソースへの割り当てに関する詳細については、『Kofax TotalAgility のヘルプ』の「ルールの適用によるリソースの割り当て」トピックを参照してください。

ロールとリソース グループ

タスクを実行する個人が不明であるか、定期的に変更される可能性が高い場合には、ロールを使用します。ロールには、リソース グループよりも柔軟性があります。

- ロールでは、特定の個人の名前を入力する必要がありません。構成時には、タスクを実行する実際の人の名前は分からないかもしれませんが、必要なロールは分かります。
- リソース グループはシステム全体で定義されますが、ロールに割り当てられる人はジョブごとに変わる可能性があります。
- リソース グループとロールの両方が、ビジネス プロセスのすべてのバージョンに関連付けられています。ただし、ロールのプロパティを変更する場合には、プロセスを再リリースする必要はありません。これは、プロセス マップのバージョン番号が増えないことを意味します。

外部リソース

組織の一部ではなくても、プロセス内のアクティビティを完了するために必要なリソース (例: ローン申請者) には、外部リソースを変動ロールと組み合わせて使用します。これらのユーザーはプロセスに参加することができますが、TotalAgility Workspace へのアクセスが制限されており、TotalAgility Designer へのアクセス権がありません。

ジョブまたはケースを作成したり、ジョブまたはケースの進行状況を表示したりするために外部リソースが必要な場合は、制限ユーザー ライセンスを用いて制限ユーザー セッションを使用します。これにより、非 Kofax TotalAgility ユーザーにアクセス権を追加できますが、引き続き制限および管理の対象になります。

プロセスの構築

プロセスは、アクティビティの編成です。記録システムではないため、プロセス変数を使用して、意思決定、検索、またはノードへの受け渡しに不要な情報を保存しないでください。

ケースには、複数の情報源を使用するさまざまな部門が関与するプロセスをいくつか組み込むことができます。たとえば、異議申し立ての処理は、さまざまな文書とフォーム、および異議申し立ての登録、法廷セッションの設定、医療記録の確認、不服申し立ての撤回など、多数のプロセスで構成できます。これらのプロセスは互いに独立して実行できますが、それらはすべて1つのケースに関連するものです。

Kofax TotalAgility は、いつでも同じプロセスの複数のバージョンをサポートするため、ジョブは開始されたときと同じバージョンのものを使用して完了します。

- 組み込みプロセス、またはサブジョブとして使用されているプロセスを更新するときは注意してください。
親ジョブでは最新バージョンのサブジョブまたは最新バージョンの埋め込みプロセスを使用するため、アクティブな親ジョブにこの変更を加えるかどうかを検討してください。そうでない場合は、プロセスのコピーを作成し、そのコピーを今後使用します。
- 請求書の承認、発注処理、人事処理、共有など、プロセスを論理的なカテゴリに分類します。
- ソリューションの全体的なカテゴリを1つにしたい場合は、カテゴリ内でカテゴリを使用します。

ケース対プロセス

プロセスは非常に再現性の高い明確なワークフローですが、ケースはそのパスでやや予測不可能です。

プロセスには、開始点、定義されたパス、および終了点があります。たとえば、休暇申請プロセスを前もって明確に定義し、定期的に行うことができます。

ケース プロセスは、ケース全体をサポートするために使用される基本「ケース プロセス」と、ケース固有の情報を収集し共有するためにすべてリンクされている複数のプロセスまたはフラグメントで構成されます。これは必ずしも論理的な開始から終了までという様式で実行されるとは限りません。実行する次のステップを決定するのはケース マネージャーの責任となり得ます。もしくは、ケースに組み込まれたロジックが次のステップを自動的に決定し、フラグメントおよびその他の基準に基づいて関連ジョブを作成できます。統合されたケース履歴は、ケース全体で時系列に維持されます。

たとえば、異議申し立てのケースの処理は、さまざまなドキュメントやフォームのほか、異議申し立ての登録、法廷セッションの設定、医療記録の確認、異議の承認などの多数のプロセスで構成されます。これらのビジネス プロセスはすべて、互いに独立して実行できますが、単一の異議申し立てのケースに関連するものです。

開始から終了までのパス全体が分かっている、一連の明確なステップを設定順に完了する必要がある場合は、プロセスを使用します。特定の基準に応じて実行時に必要なステップを決定する必要がある場合は、ケースを使用します。

ケースを使用するかプロセスを使用するかがまだ分からない場合は、プロセスを使用します。プロセスをケースまたはフラグメントにアップグレードすることは可能ですが、その逆はできません。

フラグメントとプロセス

設計時と実行時の両方でケース データ、通過ポイント、状態、イベント、およびその他の詳細に直接アクセスする必要がある場合は、フラグメントを使用します。ケース フラグメントはプロセスに似ています。ノード、データ、SLA、リソースなど、標準のプロセスの属性がすべて含まれています。ただし、フラグメントはそれが作成されたケースに依存し、他のプロセスまたはケースで再利用することはできません。例: フラグメントを使用して、ケース ドキュメント セットの一部としてドキュメントを処理します。

- 必要に応じて、ケース プロセスの正しいバージョンにフラグメントを揃えてください。
- 更新されたフラグメントを古いバージョンのケースで使用する必要がある場合、最も古いバージョンのケースで使用可能なアーティファクトのみを使用するように設計されていることを確認します。利用可能でないアーティファクトを使用しようとすると、予期しない動作やジョブの一時停止が発生する可能性があります。
- デフォルトでは、新規フラグメントはケースの最新バージョンとのみ互換性があり、ベースバージョンのアーティファクトのみを使用できる自動チェックは行われませんことに注意してください。これは、フラグメントをサブ プロセスとして使用する場合に特に当てはまります。

特定のケースから独立して使用する必要がある場合は、プロセスを使用します。ケース データには直接アクセスできないため、データをプロセスに渡すか、リアルタイムで取得する必要があります。例: 顧客の希望する通信チャンネルに電子メールを送信し、ケースの一部として記録します。

一般的なプラクティス

このセクションでは、最適な結果を得るために従う必要がある一般的なプラクティスについて説明します。

- プロセスの論理グループ化およびメタデータ検索には作業タイプを使用することをお勧めします。作業タイプでは、各項目を操作して主要な情報にアクセスする必要なく、適切な情報を表示する作業キューとジョブのリストを作成できます。ケース内のすべてのフラグメントでこの情報が異なるようにする場合は、メタデータの [範囲] プロパティを [はい] に設定してください。
- 作業キューに多くのタイプが含まれる可能性がある場合、追加情報を表示する手段としてサポート情報を使用することを検討してください。これは、画面スペースが非常に限られているモバイル デバイスで特に有効です。

- 作業タイプのクエリを作成します。たとえば、ローンの申請には、クエリ エディタ内でジョブ情報 (顧客名と口座番号) を表示およびフィルタ処理できる柔軟性があります。

作業キュー

作業キュー							
すべてのアクティビティ (デフォルト) リフレッシュ クエリの編集							
	アクティビティ名	プロセス	優先度	期限	割り当て先	SLA (...)	アクション
マイクエリ	アクティビティ 1	新しいプロセス	1	2020/06/14	全員	●	選択 ▼
共有クエリ	アクティビティ 1	新しいプロセス	1	2020/06/14	全員	●	選択 ▼
アラート	アクティビティ 1	新しいプロセス	1	2020/06/14	全員	●	選択 ▼
キャプチャ スタートバック	アクティビティ 1	新しいプロセス	1	2020/06/14	全員	●	選択 ▼
すべてのアクティビティ (デフォルト)	アクティビティ 1	新しいプロセス	1	2020/06/14	全員	●	選択 ▼
マイ アクティビティ	アクティビティ 1	新しいプロセス	1	2020/06/14	全員	●	選択 ▼
マイ ワーク	アクティビティ 1	新しいプロセス	1	2020/06/14	全員	●	選択 ▼
今日が期限のすべてのワーク	アクティビティ 1	新しいプロセス	1	2020/06/14	全員	●	選択 ▼
今日が期限のマイ ワーク	アクティビティ 1	新しいプロセス	1	2020/06/14	全員	●	選択 ▼
今週が期限のすべてのワーク	アクティビティ 1	新しいプロセス	1	2020/06/14	全員	●	選択 ▼
今週が期限のマイ ワーク	アクティビティ 1	新しいプロセス	1	2020/06/14	全員	●	選択 ▼
自分とマイ グループ (全員を除く)	アクティビティ 1	新しいプロセス	1	2020/06/14	全員	●	選択 ▼
自分とマイ グループのワーク	アクティビティ 1	新しいプロセス	1	2020/06/14	全員	●	選択 ▼
自分と部下のワーク	アクティビティ 1	新しいプロセス	1	2020/06/14	全員	●	選択 ▼

クエリ エディタ - メタデータ フィルタ

メタデータ

フィルタ演算子: AND

メタフィールド: なし | 演算子: > | 値:

作成 更新 リセット

フィールド	タイプ	演算子	値

- 遵守するサービス レベル契約がある場合は、プロセス レベルとアクティビティ レベルの両方で SLA 機能を使用します。これにより、より有益な作業キューとジョブ リストが提供され、ジョブが目標を達成できない可能性がある場合に視覚的な表示が行われます。

作業キュー - ローン申請キュー

作業キュー							
すべてのアクティビティ (デフォルト) リフレッシュ クエリの編集							
	アクティビティ名	プロセス	優先度	期限	割り当て先	SLA (...)	アクション
マイクエリ	アクティビティ 1	新しいプロセス	1	2020/06/14	全員	●	選択 ▼
共有クエリ	アクティビティ 1	新しいプロセス	1	2020/06/14	全員	●	選択 ▼
アラート	アクティビティ 1	新しいプロセス	1	2020/06/14	全員	●	選択 ▼
キャプチャ スタートバック	アクティビティ 1	新しいプロセス	1	2020/06/14	全員	●	選択 ▼
すべてのアクティビティ (デフォルト)	アクティビティ 1	新しいプロセス	1	2020/06/14	全員	●	選択 ▼
マイ アクティビティ	アクティビティ 1	新しいプロセス	1	2020/06/14	全員	●	選択 ▼
マイ ワーク	アクティビティ 1	新しいプロセス	1	2020/06/14	全員	●	選択 ▼
今日が期限のすべてのワーク	アクティビティ 1	新しいプロセス	1	2020/06/14	全員	●	選択 ▼
今日が期限のマイ ワーク	アクティビティ 1	新しいプロセス	1	2020/06/14	全員	●	選択 ▼
今週が期限のすべてのワーク	アクティビティ 1	新しいプロセス	1	2020/06/14	全員	●	選択 ▼
今週が期限のマイ ワーク	アクティビティ 1	新しいプロセス	1	2020/06/14	全員	●	選択 ▼
自分とマイ グループ (全員を除く)	アクティビティ 1	新しいプロセス	1	2020/06/14	全員	●	選択 ▼
自分とマイ グループのワーク	アクティビティ 1	新しいプロセス	1	2020/06/14	全員	●	選択 ▼
自分と部下のワーク	アクティビティ 1	新しいプロセス	1	2020/06/14	全員	●	選択 ▼

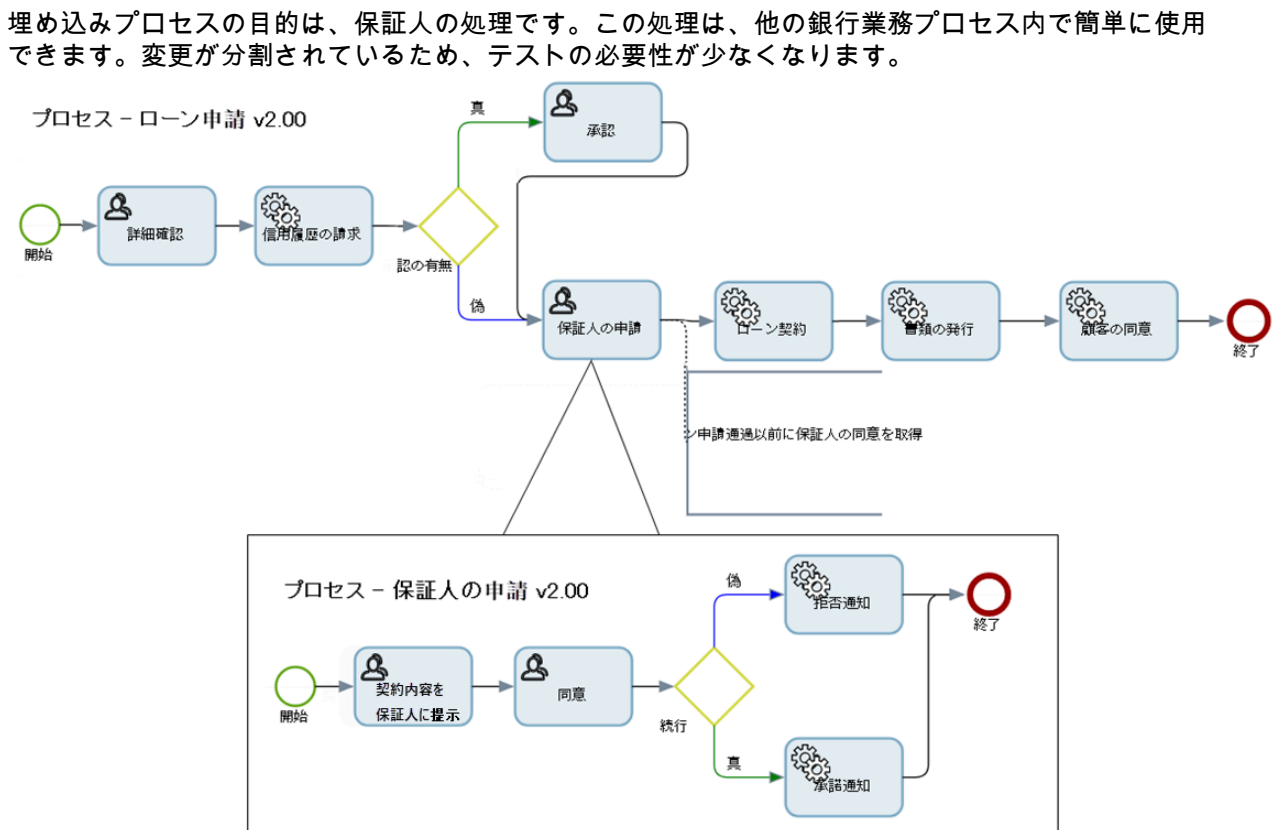
たとえば、作業キュー - ローン申請キューの SLA (アクティビティ) 列を見ると、Bob Smith と Jane Cameron のローン申請が期限に間に合わないことがわかります。

- こういったアクティビティが期限切れになることを事前に把握することで、多くの費用をかけずに修正措置を取ることができます。例外とトリガーを使用して自己管理するプロセスを設計します。たとえば、作業を再割り当てできるように、アクティビティの期限が切れる前にトリガーを起動するように構成し、アクティビティの期限が切れた場合は例外を発生させるようにします。
- ソリューション内のプロセスで迅速な応答が求められる場合、ソリューション設計内で同期処理、アクティビティの完了と進捗、新しいジョブの作成と進捗などの機能を検討してください。

プロセス設計ガイドライン

プロセスを設計するときは、次のガイドラインを使用してください。

- 変数の使用と目的を適切に反映するように、変数には明確な表示名を使用します。
- サブジョブと埋め込みプロセスを使用して、アクティビティの論理グループを作成します。メンテナンスを容易にするために、プロセスフローを理解しやすい状態に保ちます。再配置オプションを使用して、プロセスを直線的に表示し、読みやすくします。
- 変数の使用と目的を反映するように、変数には明確な表示名を使用します。
- 意図を明確にするためにアクティビティを色分けします。たとえば、次のローン申請プロセスでは、プロセスを一時停止する可能性のある手動アクティビティにカスタム カラーを使用しています。カスタム カラーは、統合ポイントと埋め込みプロセスにも使用されています。



- 注釈を使用して、設計のコンテキストまたは説明を追加します。
- 本番環境にリリースした後には作業タイプを変更し、フィールドを追加および削除する場合、フィールドを削除して別のタイプのフィールドを追加しないでください。データがデータベースに特定の位置で文字列として保持され、使用時に変換されるため、ジョブが既に存在する場合は、問題を引き起こす可能性があります。たとえば、名前 (文字列) フィールドが位置 1 にあり、年齢 (数値) フィールドが位置 2 にあり、複数のジョブが既に作成されている場合、名前フィールドを削除すると、実行時の問題が発生します。システムは、アルファベットのテキストが含まれている可能性がある既存のジョブを数値に変換しようとするため、変更は失敗します。

- プロセス設計をより明確にするための視覚的な補助として、異なるタイプの開始ノードと終了ノードを使用します。

注 すべての終了ノードが現在のジョブを終了するわけではないため、終了ノードでパスを終了し、同時に他のパスの処理を継続することができます。これらのノードの詳細については、『Kofax TotalAgility のヘルプ』の「終了ノードのイベント タイプ」トピックを参照してください。

- 依存先のない並列パスを持つマップを設計するときは、完了しない終了ノードを使用して、マップをより読みやすくします。不要な同期ポイントまたは依存先のある構成を避けてください。次の図の例を参照してください。

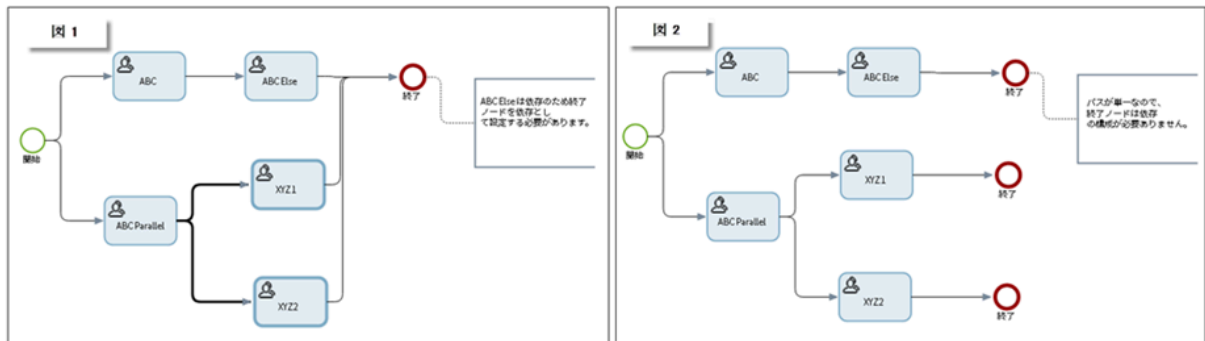


図 2 では、「XYZ1」および「XYZ2」アクティビティが完了したときは、ジョブは完了しません。「ABC Else」アクティビティが完了したときのみ、ジョブが完了します。図 1 にあるような依存先は終了ノードに構成されておらず、複数の収束ラインも存在しません。こうすることで図 2 のマップが読みやすくなります。

- ケースフラグメント変数にケース変数と同じ名前を付けないようにします。同じ名前にすると、その変数がフラグメント内で使用されているとシステムが誤って認識するため、ケース変数を削除するときに問題が発生します。また、マップ内にどの変数が使用されているかを示すインジケータはないため、マップ構成を表示するときに混乱につながる可能性があります。スコープ規則（プロセス、ケース、サーバー）により、実行時にプロセス変数が常に使用されます。

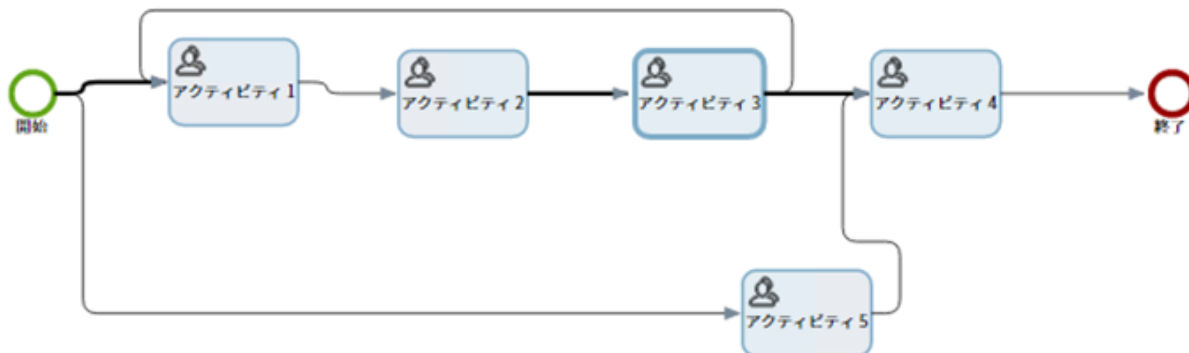
プロセス パフォーマンス ガイドライン

プロセスのパフォーマンスを向上させるには、次のガイドラインを使用します。

- 内部および外部のプロセス間通信には、プロセス イベントを使用します。例: ローン申請の処理は、ローン契約に署名した後にのみ続行できます。
- スリープ ノードとポーリング プロセスはパフォーマンスにマイナスの影響を与えるため、お勧めしません。外部システムを呼び出す必要がある場合に、非常に短い待機でのみ使用してください。プロセス イベントの構成と使用の詳細については、『Kofax TotalAgility のヘルプ』の「プロセス イベントの作成」トピックを参照してください。
- 長時間実行されているアクティビティは、定義されたタイムアウト期間を超えると TotalAgility によってキャンセルされます。ただし、特定の状況では、RPA ロボットなど、一部のアクティビティは、長い時間実行される場合があります。そのため、展開する前に、特定のアクティビティに対してアクティビティの想定される実行時間を指定する適切なタイムアウトが定義されていることを確認してください。

システム設定は変更しないでください。変更すると、アクティビティが必要以上に長い時間実行され、マシン リソースを消費することになり、パフォーマンスに影響を与えます。

- 監査目的でジョブ履歴を追跡する必要がない場合は、プロセス内の「履歴の記録」設定をクリアします ([プロセスのプロパティ] > [履歴、レポート、および実行] タブ)。これは、データベース サイズの管理とパフォーマンスの向上に役立ちます。
- 条件と依存先のある同じノード (または一連のノード) を繰り返すプロセスに注意してください。それらのノードは依存先の完了を待ちますが、依存先が条件によってはアクティブになることがない場合、マップの進行が止まることとなります。例: アクティビティ 3 をアクティビティ 4 の依存先に設定すると、アクティビティ 4 が保留中にならなくなります。



- 同期ノードを不必要に使用しないでください。同期ノードは、プロセス内の複数のパスをマージおよび収束し、システムに負荷を増やし、データベースのパフォーマンスを低下させる可能性があります。同期ノードを出入りする複数のパスを持つプロセスを設計し、同期ノードが完了するためにすべてのパスを完了する必要がある場合は、依存ノードを使用します。
- 作業タイプ固有のキューではなく一般的な作業キューまたはジョブ リスト フォームを使用する場合は、「サポート情報」機能を使用してジョブ 関連情報を表示します。

アクティビティ名	優先度	サポート情報	期限	割り当て先	SLA (ジョブ)	SLA (アクティビティ)
アクティビティ 1	1	Bob -C46902R	2020/06/23 22:44:07	全員	●	●
アクティビティ 1	1	Sam -C46902R	2020/06/23 22:44:27	全員	●	●
アクティビティ 1	1	John Smith -B39756h	2020/06/23 22:49:36	全員	●	●

プロセスのテスト

プロセスをテストするときは、次のプラクティスに従うことをお勧めします。

- 本番に移行する前に、プロセスのすべてのパスをテストします。
- ソリューションのユーザー インターフェイスを作成する前に、デフォルトで使用可能なフォームを最初に使用して、プロセスが期待どおりに実行されることを確認します。
- プロセスの単一部分をテストする場合は、新しいジョブを作成して多くの手順を実行するのではなく、ジョブ ビューア内で再起動機能を使用します。たとえば、ビジネス ルールに変更を加えた場合、新しいジョブを作成するのではなく、ビジネス ルール ノードでプロセスを再起動します。

注 場合によっては、変更には新しいジョブまたはジョブのアップグレードが必要になります。

- テストの範囲を理解します。小さな変更は、必ずしもマップ全体を再テストする必要があることを意味しません。

- サブプロセスまたは埋め込みプロセスを使用する場合、最新バージョンのみが使用されることに注意してください。プロセスを切り離して実行およびテストできることを確認します。たとえば、コントラクツの初期化データが変更された場合、サブプロセスが呼び出されると問題が生じる可能性があります。サブプロセスの特定のバージョンにプロセスを合わせる必要がある場合は、必要な更新のコピーを作成し、必要に応じてそのコピーを使用します。または、ケースとフラグメントを使用することで、バージョンを揃えます。
- ビジネスルールを使用している場合は、ビジネスルールをテストする組み込み機能を使用して、さまざまな入力値を使用してテストします。

機能アクセスの設定

アクセス タイプの範囲を設定して、ライブ ジョブで実行できる機能を制御できます。

デフォルトでは、誰でもジョブの作成、一時停止、強制終了、再開、ホールド、ジョブの詳細を表示、スキャン/VRS プロファイルの変更、分割プロファイルの変更、ドキュメント セットの更新を行うことができます。

デフォルトでは、ライブ ジョブでカスタマイズ バージョンのプロセスを作成できません。

リソースのスキルと役割を考慮して、機能ごとのアクセスをより具体的なリソースに割り当てることをお勧めします。たとえば、どのリソースがジョブを作成できるか、どのリソースが再起動または強制終了できるか (それらのリソースがジョブの強制終了または再起動の結果を認識できるようにする)、またはどのリソースにジョブの詳細の表示のみが許可されるべきかを検討します。

フォームの構築

フォームを作成するときは、次のプラクティスを利用してください。

フォームの使用の適合性

Kofax TotalAgility Form Designer は、Visual Studio に代わるものではありません。ただし、ドラッグアンドドロップ インターフェイスを使用してフォームを簡単に作成できるように設計されています。意図した開発環境としてフォームを選択する場合、Kofax TotalAgility を使用して何を構築できるか、およびその制限事項を理解することが重要です。

- フォームをシンプルに保ち、Kofax TotalAgility の強みを活用します。Kofax TotalAgility Form Designer の具体的な強みは、新しいジョブ/ケースの作成、作業キュー、アクティビティの進捗 (キャプチャ機能あり/なし) などのコア機能に使用されるフォームを自動で生成できることです。
- キャプチャ関連の機能を使用するソリューションでは、コア キャプチャ関連アクティビティ用の Kofax TotalAgility フォームをすぐに使用できます。これらのフォームを使用することで開発時間を大幅に短縮できます。

設計ガイドライン

フォームを設計するときは、次のガイドラインを使用します。

- マージンと非表示コントロールに頼らずに、セルと列を使用して必要なレイアウトを作成し、フォームを読みやすくします。それぞれのフォームを理解しやすいものにする必要があります。チームの新しいメンバーがロジックに従って、簡単にフォームをメンテナンスできる必要があります。

- コントロールとアクション用にそれらの使用と目的を反映する明確な表示名を使用します。一度作成した名前を変更することはできません。
- メンテナンスの複雑さを低減しコストを削減するには、次の事項を実施します。
 - 10 ~ 20 を超えるアクションが発生するイベントがないことを確認します。
 - イベントに過剰な条件付きロジックは含まれていません。
 - フォーム上のコントロールの数を最小限に維持します。
- すべてのフォームは、デフォルトのメニューとヘッダーを使用するように設定されています。不要なメニューとヘッダーを更新したり削除したりします。例：
 - デフォルトのヘッダーには未読のリソース メモへのリンクがあり、API 呼び出しを行います。リソース メモを使用していない場合は、ヘッダーを変更するか、既存のヘッダーを更新します。
 - ユーザーがアクティビティをキャンセルせずにページから移動できないように、アクティビティベースのフォームからメニューを削除します。メニューとヘッダーの更新は、ドキュメントのロックスも保持するため、キャプチャ アクティビティに特に関連しています。
 - メニューの削除は、ログオンまたはログオフ機能を含むフォームにも適用されます。これらを更新しないと、有効なセッションを利用せずにデータまたは機能にアクセスできます。
- デスクトップ フォームと電話フォームでは、ナビゲーションとフォーム フローが大きく異なるのが一般的です。したがって、同じフォーム設定を使用し、同じフォーム内にデスクトップ、電話、およびタブレットのバージョンを含めても意味がありません。ソリューションの構造を早期に決定してから、適切な設計要素を選択してください。
- 他のシステム、またはアクティビティを完了するために必要なデータを表示する必要がある場合は、マルチビュー アクティビティ フォームを使用します。
- 電話フォーム、特に MFP フォームを使用する MFP デバイスで現在のビューを利用して、リダイレクトを大幅に改善します。これにより、Extjs ライブラリをリロードするデバイス上のリソース不足を回避し、読み込み時間を短縮できます。他のモバイル デバイスにも適用できますが、それらに同じレベルの改善は見られません。

Kofax TotalAgility には補正トランザクションのサポートが内蔵されていないことに留意してください。必要に応じて、実装プロジェクトの一部として作成します。

キャプチャと非キャプチャの組み合わせ利用

非キャプチャ データでキャプチャ フォームを強化する場合は、これらのプラクティスに従ってください。

- 可能であれば、非キャプチャ データをアクティビティ画面または初期化画面に保存します。そうしないと、予期せぬ動作が発生する場合があります。たとえば、非表示タブの必須フィールドがユーザーに表示されない場合があります。
- ドキュメントに対してフィールドが必要な場合は、フォーム上でフィールドを定義するのではなく、ドキュメントに対してフィールドを定義することを検討してください。たとえば、非キャプチャ フィールドをキャプチャ フォームに追加して必須にすると、ドキュメント フォームが読み込まれず期待される検証が実行されないため、フォルダの検証時に予期せぬ結果が生じる可能性があります。

フォームの再生成

TotalAgility は、いつでも同じプロセスの複数のバージョンをサポートするため、ジョブは開始されたときと同じバージョンのものを使用して完了します。

フォームの再生成機能を使用して、さまざまなバージョンのプロセスとさまざまなバージョンの関連付けられたフォームに対応できます。

再生成機能には多彩なオプションがあり、さまざまな出力が得られます。

- フォームの新規バージョンの作成、または上書き: 現在のバージョンのフォームを正常に実行する必要があるライブ ジョブがある場合は、新しいフォームを作成します。
- 新しいフォームの保存、またはリリース: フォームは利用可能な場合にのみ、リリースします。
 - 対応するプロセスがリリースされています。
 - フォームにさらに変更を加える必要はありません。
 - 新規ジョブで使用できます。

フォーム/ナビゲーション/ヘッダーの再利用

フォーム、ナビゲーション、ヘッダーを再利用することで、再作業を避けてメンテナンスを容易にします。

- 最小限のフィールドを持つアクティビティ、または少数のフィールドを持つドキュメントやフォルダの場合、フォームを常に生成する必要はありません。フォームを生成する前に、フォームに何が必要かを確認し、カスタマイズまたはカスタム ロジックが必要かどうかを確認します。フォームを生成すると、実行時にシステムが自動的にユーザー インターフェイスを作成します。これにより、フォーム数が削減され、その後のメンテナンスが軽減されます。
- Workspace ナビゲーションをソリューションのデフォルトとして想定しないでください。Workspace とそのデフォルトのナビゲーションを使用すると、通常必要とされない管理機能を含むすべての機能への即時アクセスが提供されます。ソリューションのコンテキストに必要な正しいナビゲーションを設計してください。
- アクティビティが「取得済み」状態のままユーザーがブラウジングしたり作業を終了したりせずに、アクティビティを完了またはキャンセルできるように、アクティビティ実行フォームからナビゲーション メニューを削除することを検討してください。
- 複数のソリューションで作業キューや管理ページなどの同じ機能が必要な場合は、これらのフォームを一度作成して、ソリューション間で共有します。サイトに関連付けられたテーマを使用して、フォームのスタイルを変えることができます。

セキュリティ

どのユーザーがフォームとメニュー項目に完全にアクセスできるようにするか、およびフォーム上のすべてのコントロールをすべてのユーザーが使用できるようにするかを検討します。

特定のリソースへのアクセスを許可または拒否するために、フォームおよびナビゲーション メニュー項目でセキュリティを使用できます。デフォルトでは、すべてのユーザーがアクセスできます。

さらにセキュリティを細かく設定する場合は、「セキュリティ トークン」機能を使用します。フォーム上の個別のコントロールまたはセルにトークンを割り当てます。これらのトークンを持つロールに属するリソースのみがコントロールを表示できます。

注 セキュリティ トークンを持つフォームのレンダリングでは、現在のリソースのロールを決定するために追加の呼び出しを行ってから表示するコントロールを決定する必要があるため、オーバーヘッドが生じます。

フォームの保守性

フォームを作成または変更するときは、将来の保守性を考慮します。保守性は、ソリューションが成功するための重要な要素になります。

すべての入出力の数、名前、タイプなど、アクティビティのコントラクト (入力および出力) がまったく同じである場合は、それらのアクティビティ全体でフォームを再利用します。これにより、フォーム数が削減され、その後のメンテナンスが軽減されます。同じフォームを使用するには、次のプロセスまたはアクティビティのプロパティを設定します。

ジョブ作成フォーム ([プロセスのプロパティ] > [アクセス] タブ)

関連ファイルのパス ([アクティビティ プロパティ] > [構成] タブ)

フォーム ロード イベント

「Loaded」、「BeforeRender」および「AfterRender」イベントを使用する場合、それらが次の順序で起動されることを確認してください。

1. ロード済み
2. BeforeRender
3. AfterRender

Loaded イベントは、最も一般的に使用されるイベントです。通常、コントロールまたはフォーム変数を初期化する同じページ、.NET、DB クエリなどのアクションがこのコントロールに追加されます。この時点ではフォーム コントロールを DOM に読み込めないため、DOM からフォーム コントロールにアクセスしようとするアクションを配置しないでください (例: Javascript アクション)。

BeforeRender イベントは、フォームがレンダリングされる直前に発生します。たとえば、BeforeRender イベントを使用して、レンダリングの前にドキュメントの先頭に css を挿入します。Loaded イベントにより、ここで実行される Javascript アクションでは、DOM からフォーム コントロールへのアクセスを試行しないでください。それらのコントロールはこの時点では使用できないためです。[JavaScript アクションによる DOM 要素へのアクセス](#)を参照してください。

フォームは DOM に完全に読み込まれているという保証があるため、**AfterRender** イベントを使用して DOM 要素にアクセスできます。これは、EventListeners を DOM 要素に追加していくつかの非標準機能を実現するときに特に役立ちます。

注 キャプチャ アクティビティ フォームは、レンダリング方法が他のフォーム タイプとは異なります。これは、ThinClient コントロールによるものです。ThinClient コントロールでは、onLoad アクションが完全にレンダリングされる前に完了する必要があります。そのため、BeforeRender イベントまたは AfterRender イベントが包含フォームでトリガーされた場合でも、フォーム自体が実際に必要な状態にレンダリングされていない可能性があります。

JavaScript アクションによる DOM 要素へのアクセス

DOM クエリを使用して Javascript アクションでコントロールの値にアクセスするには、コントロールが完全にレンダリングされている必要があります。

たとえば、Javascript アクションの次の行を使用します。

```
var controlValue = document.getElementsByName("textbox1")[0].value;
```

この Javascript アクションを「Loaded」イベントまたは「BeforeRender」イベントのいずれかに関連付けると、フォームが完全にレンダリングされず、DOM 要素が利用できないため、失敗します。

ただし、アクションを「AfterRender」イベントに関連付けて、フォームが完全にレンダリングされると、コードが期待どおりに実行されます。フォームが完全にレンダリングされると、「TextChanged」や「Clicked」などのコントロール イベントにアクションを関連付けることにも成功します。

JavaScript アクション以外のアクションへの関連付け

同じページや .NET アクションなどの非 JavaScript アクションは、JavaScript アクションと同じ制限を受けません。これらのアクションは、DOM に直接アクセスしてコントロール データを取得または設定しようとしません。たとえば、同じページ アクションは、「Loaded」、「BeforeRender」または「AfterRender」イベントに関連付けられているかどうかに関係なく、あるコントロールから別のコントロールにデータを正常にマッピングします。

.NET アクションなどのサーバー側のアクションは、イベントに対して順々に定義されると、サーバー上で同期的に (次々に) 実行されます。ただし、複数のイベントが発生した場合、アクション自体がサーバー上で同期的に実行されていても、サーバー側の各アクション セットはクライアントから非同期的に実行されます。したがって、サーバー側のアクションを「Loaded」、「BeforeRender」および「AfterRender」イベントに関連付けるときは、あるイベントに関連付けられたサーバー側のアクションが別のイベントに関連付けられたアクションの前に完了するという保証はないため、注意してください。完了の順序は予想どおりではない場合があります。

フォームのテスト

フォームを入念にテストします。

- 「以下のときに再起動」機能を使用して、作成されるジョブの数を最小限にします。
- ビジネスルールとその他の統合項目は、分離してテストしてから、フォームに統合します。
- 複数のアクションが関係する場合は、いくつかのアクションを非アクティブ化またはアクティブ化して、問題を特定し、切り離します。

フォームのリリース バージョンまたは開発バージョンがない場合、テスト目的でフォームを継続的にリリースすると、ソリューション内の最終バージョン番号が増加することに注意してください。

作業キューから検証アクティビティをスキップしないようにする

TotalAgility 作業キュー フォームから、アクティビティを一切実行せずにアクティビティを自動完了できます。そのためには、アクティビティの [アクション] 列で [アクティビティの完了] をクリックします。ジョブに無効なドキュメントが残っている場合でも、検証アクティビティは完了します。

ユーザーがアクティビティを自動完了できないようにし、アクティビティが完了する前に実際に作業されるようにするには、作業キューからオートコンプリート機能を無効にします。これを行うには、作業キュー フォームを開きます。作業キュー コントロールのプロパティ パネルで、[アクション] の [構成] タブから [オートコンプリートを許可] をオフにします。

ビジネス ルール

ビジネス ルールは、カスタム コードを必要とせずに複雑なビジネス ロジックを実装する手段です。その主な目的は、提供された入力に基づいて結果を決定することです。たとえば、合意された契約条件に応じて、ローンの金利を決定します。ビジネス ルールは手動アクティビティを含むことはできません。ビジネス ルールは直接的な処理を目的としたものであるため、記録システムやその他のそのようなアクティビティの更新には使用しないでください。この機能が必要な場合は、同期プロセスを使用してください。

ビジネス ルールを設計するときはプロセス設計ガイド ラインに従い、不必要な入力と出力を避け、必要に応じてサブルールを使用します。

ビジネス ルールの保守性

本番への展開後にコントラクト (入力および出力) が変更されないようにします。そうしないと、ビジネス ルールが利用されている場所に破損が起きます。ビジネス ルールのコントラクトを変更する必要がある場合は、ルールが本番環境において利用されていないことを確認するか、ルールのコピーを作成して、それを使用し新しい ルールを作成します。古いルールを使用しない場合は削除します。

ビジネス ルールのテスト

ビジネス ルールのプロパティ内のテスト機能を使用して、サンプル入力を使用して結果の出力を確認できます。出力が期待どおりでない場合は、フローを手動で調べて間違っている箇所を判断します。または、さまざまなポイントで出力変数を使用して式ノードで値を更新することでトレースを追加できます。

トレースを追加するにはルールをリリースする必要があるため、ビジネス ルールのバージョン番号が増加することに注意してください。

詳細については、[プロセスのテスト](#)を参照してください。

ジョブ アップグレード プログラム

プロセス設計を変更して、すべての新規ジョブおよびライブ ジョブにすぐに反映させる必要がある場合は、ジョブ アップグレード プログラムを使用します。たとえば、規則が変更されると、SLA の変更が必要になります。

ジョブ アップグレード プログラムは必要な場合にのみ使用してください。予期しない動作を引き起こす可能性があるため、使用する場合は注意してください。たとえば、新規アクティビティ内で新しい変数が使用されると、アップグレードされたジョブにはデフォルト値のみが含まれる場合があります。

ソフトウェアのアップグレードに適用するのと同じルールをアップグレードにも適用します。たとえば、データベースをバックアップし、適切な回帰テストを実行します。回帰テストでは、ライブ データベースのバックアップを使用し、テストデータとしてアップグレードされる実際のジョブを使用します。

データ管理

エンティティは、会社、顧客、住所などの詳細など、TotalAgility でデータを定義し保存するためのライトウェイト メカニズムを提供します。技術に詳しくないユーザー、またはデータベースがないときやデータベースのスキルがないときに適しています。

実際のデータベース モデルにあるのと同じレベルのスケーラビリティやセキュリティを提供するものではありません。システム間で実際のインスタンス データを移動することはできません。

基本的な検索のみを実行できますが、データのレポートはできません。

この機能は、複数のテーブルにわたる表示、外部キー、集中的な検索など、データの高い整合性を必要とする複雑なデータ構造の処理には適していません。

分散型のアップグレード可能性

多くの顧客に配布されるソリューションを構築する場合、複数の顧客にソリューションのアップグレード バージョンを送信できるように、アップグレード可能性を考慮してソリューションを設計します (例: 2、3、4)。

すべてのパッケージが、次の 2 つの部分で構成されている必要があります。

- コア: 顧客が変更できない保護されたアイテム。
- カスタム: 顧客が変更できるアイテム。

新しい顧客は、両方のパッケージを含むパッケージをインポートする必要があります。

これの完全なコントロールが可能なので、リリース サイクルに基づいてコア パッケージを更新し、顧客に送信できます。

カスタム パッケージでは、顧客がこれらを必要とするかどうかを決定する必要があります。コアの変更の見込みに基づいて必要になる可能性があるため、常に新しいアイテムを用意してください。アイテムが変更された場合も、更新されたアイテムが必要かどうかは顧客の判断です。ただし、特定のルールに従う必要があります。たとえば、カスタマイズでは、コア アイテムとカスタム アイテム間のコントラクト (初期化データ) を変更することができません。これは、新しいジョブやアクティビティの作成フォームなどのフォームにも関連しています。

カスタマイズ ポイントを使用してプロセスを設計します。

第7章

トラブルシューティング

この章では、ビジネス プロセス管理、キャプチャ、Transformation Server、および VRS に関するいくつかの問題とそれらの回避方法または解決方法を説明します。

インストールのトラブルシューティングについては、『Kofax TotalAgility インストール ガイド』を参照してください。

ビジネス プロセス管理

このセクションでは、フォーム、プロセス、ビジネス ルール、およびパフォーマンスに関連する問題について説明します。また、一般的な問題、およびそれらを回避または解決できる可能性のある方法を説明します。

フォーム

- フォームが期待どおりに表示されない場合は、「プレビュー」を使用してトラブルシューティングを行うか、フォームを展開して、展開された環境でフォームの機能をテストします。
- フォームが期待どおりに機能しない場合、たとえば、コントロールが正しく入力されない、またはアクションが期待どおりに実行されない場合に、動作を切り分けるには、デバッグするアクションに依存していない他のすべてのアクションを無効にすることをお勧めします。
- フォーム変数など、画面上に表示されていないデータを表示する場合は、デバッグ パネルを作成し、「同じページ」アクションを使用して必要なデータをコントロールに入力します。トラブルシューティングが完了したら、パネルを削除します。

プロセス

- ソリューションの一部として同期プロセスが必要な場合は、最初に非同期としてプロセスを作成して、動作が正しいことを確認します。同期プロセスが作成されたら、次の手順を実行してプロセスをデバッグします。
 - 追加のエラー情報については、イベント ログを確認してください。
 - レコード履歴をオンにして、ジョブ ビューアで取得したパスを表示し、変数の最終値を確認します。
 - これをさらに細分化する必要がある場合は、さまざまなブレイクポイントで終了ノードに決定や分岐ルール (XOR) を追加して、各ポイントの変数値を確認します。
 - 問題が分割できたら、さらに切り分けてテストするために、テスト プロセスにノードをコピーします。

- 非同期プロセスをデバッグするには、次の手順を実行します。
 - 追加のエラー情報については、ジョブのメモを確認します。
 - レコード履歴をオンにして、ジョブ ビューアで取得したパスを表示し、変数の最終値を確認します。
 - ブレークポイントをシミュレートし、変数値を調査および設定するには、手動アクティビティを追加します。
 - 「以下のときに再起動」機能を使用して、ブレークポイントにジャンプします。
- 作業キューに作業が期待どおりに表示されない場合は、Workspace 内でジョブ ビューアを使用して、プロセス内の現在のジョブの場所およびアクティビティに割り当てられたリソースを確認します。また、以下の状態であることを確認します。
 - アクティビティは非依存パス上にあるために削除されておらず、ループに特別な注意が払われている。
 - アクティビティがグループまたはロールに割り当てられている場合、メンバーは想定どおりになっている。
 - アクティビティが保留中になるのを妨げる事前条件がない。
 - アクティビティが保留中になるのを妨げる依存先はなく、各依存先パスを同一回数実行する必要があるためにループに特別な注意が払われている。
 - アクティビティがまだ取得されていない。これは、Workspace の [取得済みアクティビティのリスト] 画面でも確認できます。
 - ユーザーは除外されたリソースではない。
 - いずれの終了ルールも満たされていない。

ビジネス ルール

TotalAgility Designer 内でサンプル入力および出力を使用してビジネス ルールをテストします。それでも問題が解決しない場合は、「プロセス」の「同期」セクションを参照して、さらにトラブルシューティングを進めます。

パフォーマンス

プロセスの一般的なパフォーマンスが良好でない場合は、プロセス履歴で実行時間を確認し、実行が遅いかどうかを判断します。ジョブ履歴用のデータベース内の [保留中期間 (秒)] パラメータを使用して、ポーリングによりアクティビティの保留中期間が想定以上に長いかどうかを判断します。

ジョブ評価、保持ポリシー、アーカイブ、監視などのシステム タスクが期待どおりに実行されない場合は、実行間隔が期待どおりであり、Core Worker サービスが開始されていることを確認してください。

全般

自動アクティビティが実行されておらず、Core Worker が実行されており、ログにエラーが書き込まれていない場合は、他のスケジュールされたアクティビティより優先されるループ アクティビティがないことを確認します。

キャプチャ

Transformation Designer と実行時の Transformation Server でドキュメントの分類または抽出が異なる場合、実際のイメージの分類または抽出が異なっている可能性が高いです。

この分析のために、次の操作を実行します。

1. プロセス内の分類または抽出の直前に一時的なダミー アクティビティを配置してワークフローが停止するようにします。
2. Repository Browser を開き、ドキュメントを見つけます。
3. コンテキスト メニューを使用して、ドキュメントをディスクにエクスポートします。

このドキュメントを Transformation Designer にロードしたら、このドキュメントと元のドキュメントの違いを分析できます。

Transformation Server

Transformation Server でアクティビティを取得できないことを分析するには、次のことを確認します。

- Transformation Server が実行されている。
- Transformation Server サービスのインストールで指定されたアカウントは、Windows ローカル ポリシーで「サービスとして実行」を設定している。
- プッシュ以外のアクティビティの処理を想定している場合、Transformation Server の構成ファイルで「EnableSynchronousCalls」が false に設定されている。
- プロセス Designer で定義された事前条件がキャプチャ アクティビティですべて満たされている。
- Transformation Server は、タスクが見つからない場合にポーリング間隔に入る。構成ファイルの「PollingTimeout」を構成することで、間隔の長さを短くできます。

注 本番システムは、ポーリング タイムアウトをサポートしていません。

VRS

スキャン/VRS プロファイルを含むプロセスまたはパッケージをインポートした後に新しいジョブのスキャン中に次のエラーが表示された場合、IIS を再起動し、インポートされたスキャン/VRS プロファイル設定がクライアントに伝搬されることを確認します。

選択された VRS プロファイルは見つかりませんでした。技術的な支援が必要な場合は、アクションを参照してください: ScanControlLoadBatchAction 。

第 8 章

PDF の取り扱い

Kofax TotalAgility では、PDF ドキュメントをインポートして、イメージ処理、分類、抽出、Document review、Validation、Verification などのキャプチャ ワークフローで処理できます。

ただし、最高のエクスペリエンスとパフォーマンスを確保するには、次のベスト プラクティスをお勧めします。

- PDF ドキュメントを処理する場合、イメージ処理アクティビティを使用して PDF ドキュメントを処理し、OCR 投げ縄などのアクションで TIFF 表現を使用できるようにします。イメージ処理アクティビティは、TIFF 表現を PDF ドキュメントに追加するだけです。PDF ドキュメントまたは PDF ドキュメント内のテキスト レイヤー情報は削除されません。
- イメージ処理アクティビティを使用して PDF ドキュメントを処理する場合、自動クロップ、自動回転、傾き補正、およびページの寸法に関連するその他の設定を変更する可能性のある処理アルゴリズムを無効にすることをお勧めします。

PDF テキスト レイヤーが抽出に使用されている場合、または イメージ処理により Transformation Designer で使用されている PDF ドキュメントに寸法の違いが生じる場合は、処理アルゴリズムを無効にすることが重要です。

- 特に変換および検証アクティビティの前に、できるだけ早い段階でプロセスにイメージ処理アクティビティを配置します。オンライン学習が変換および検証と同じイメージで発生する必要があるため、これによりオンライン学習が適切に機能することが保証されます。

ビジネス プロセスで PDF が必要な場合は、PDF 生成を使用して PDF ドキュメントを再作成します。

第 9 章

抽出グループと分類グループの設計

この章では、分類グループと抽出グループの設計のベスト プラクティスについて説明します。

分類グループおよび抽出グループ

分類グループと抽出グループは、最初に TotalAgility Designer に作成する必要があります。

分類グループには保存またはリリースする前に少なくとも 1 つの抽出グループが必要となるため、まず最初に抽出グループを作成します。

分類グループと抽出グループを作成すると、Transformation Designer でそれらを部分的に編集できます。また、TotalAgility Designer でもそれらを部分的に編集できます。

共有プロジェクト

Transformation Designer で新しいプロジェクトを作成すると、共有プロジェクトが作成されます。Kofax Transformation Modules プロジェクトをファイル システムを介して Transformation Designer にインポートする場合も同じ結果になります。

共有プロジェクトでは、ドキュメント分類とデータ抽出を 1 つの Transformation Designer プロジェクトで定義できます。このタイプのプロジェクトには、個別の分類グループまたは個別の抽出グループはありません。

注 共有プロジェクト、分類グループ、または抽出グループのプロジェクト構造の再配置は、TotalAgility Designer で行う必要があります。

分類グループおよび抽出グループと共有プロジェクト

共有プロジェクトとは対照的に、分類グループと抽出グループで作業する最大の利点は、プロジェクトの設計者が異なる分類グループまたは抽出グループで同時に作業できることです。

たとえば、分類グループ (CG) には 2 つの個別の抽出グループが含まれます。

1. EG_AB: ドキュメント タイプ A および B の処理用
2. EG_C: ドキュメント タイプ C の処理用

このようなシナリオでは、3 人のプロジェクト設計者が同時にプロジェクトの作業を行えます。例:

- プロジェクト設計者 1 は、3 つの異なるタイプのドキュメントの分類に取り組みます

- プロジェクト設計者 2 は、ドキュメント タイプ A および B の抽出に取り組みます
- プロジェクト設計者 3 は、ドキュメント タイプ C の抽出に取り組みます

ただし、グループを使用する場合の欠点は、TotalAgility Designer で分類グループのプロジェクト構造を変更すると、リンクの付いた抽出グループ プロジェクトでも、分類グループと一致するようにプロジェクト構造を変更する必要があることです。同時にそれらを編集することができないため、TotalAgility Designer で初めに分類グループを編集し、分類グループを閉じてから抽出グループを開く必要があります。その後、抽出グループ クラスを追加したり削除したり、またはクラスの Parent プロパティを使用してプロジェクト構造内で既存のクラスを移動したりすることができます。

さらに、複数の抽出グループを使用する場合、異なるグループ間で WinWrap スクリプトを複製する必要があります。

共有プロジェクトを使用しているか、分類グループと抽出グループの組み合わせを使用しているかには関係なく、TotalAgility Designer でフォーマット設定、検証構成、および検証フォーム設計を構成する必要があります。

フィールド

次のセクションでは、フィールドに関するベスト プラクティスについて説明します。

強固な命名規則

フィールドに名前を付けるときは、強固な命名規則を使用します。フィールドを識別しやすい名前にします。たとえば、PatientTelephoneNumber や InsuranceTelephoneNumber と命名するほうが、PatTelNum や InsTelNum と命名するよりも、フィールドの目的と期待値を識別しやすくなります。

フィールドの名前は Transformation Designer WinWrap スクリプトで変数名としても使用されるため、名前に目的が反映されていることが重要です。

プロジェクトが渡されたときに、新しい設計者がフィールドの名前の意味を理解できないと、スクリプトが難解になり、TotalAgility Designer でのフォーマットや検証ルールへのフィールドのマッピングも難解になります。強固で説明的な名前を付けることでこの問題を最小限に抑えます。命名規則の詳細については、『Kofax TotalAgility のヘルプ』の「命名規則」トピックを参照してください。

シーケンスとフィールド グループ

クラスは、論理シーケンスでフィールドを作成し、補助フィールドまたはダミー フィールドを含むリストを完成させてから、保存またはリリースされます。

プロジェクトが保存またはリリースされると、新しいフィールドはそのクラスのフィールド グループに追加されます。たとえば、ClassA に対して Field1 と Field2 を作成してプロジェクトを保存またはリリースすると、そのクラスの FieldGroup1 に Field1 と Field2 が追加されます。

Transformation Designer では、シーケンスが Field1、Field1a、Field2 となるように Field1a を追加して a 位置に移動すると、プロジェクトを保存またはリリースした後に、Field1a が FieldGroup2 に追加されます。次に Transformation Designer でプロジェクトを開くと、シーケンスは Field1、Field2、Field1a になります。

Transformation Designer のフィールドは、TotalAgility Designer のフィールド グループに表示される順序で表示されます。TotalAgility Designer のフィールド グループ間でフィールドを移動できないため、TotalAgility で作成する前に、フィールドとそのシーケンスを計画する必要があります。

ドキュメント フィールドの追加と削除

ドキュメント タイプのフィールドを追加したり削除したりするときに、関連するドキュメント フォームを再生成して再リリースします。アクティブなドキュメントがなく、キャプチャ アクティビティでドキュメントにアクセスできない場合は、オプションを使用して、フォームの再生成中に既存のフォームを上書きします。

更新されたタイプのアクティブ ドキュメントがある場合、フォームを再生成するときにオプションを使用して新しいフォームを作成します。この場合、新しいフォームは更新ドキュメント タイプの新しく作成されたドキュメントに使用され、以前のフォームはシステム内の既存のアクティブなドキュメントに使用されます。

第 10 章

検証とフォーマッタの実装

この章では、フォーマットと検証の実装におけるベスト プラクティスについて説明します。

フォーマット設定

Transformation Designer でフォーマット設定方法をフィールドに適用することでフォーマット設定ルールを作成できますが、TotalAgility Designer でフィールドのフォーマット設定を定義することもできます。

フォーマット設定および検証の方法は、請求書グループなどの一部のロケーターが抽出目的で使用するため、Transformation Designer で利用可能です。ただし、請求書などのドキュメント タイプの場合は、TotalAgility 抽出グループの日付フォーマッタに月の置換辞書機能がないため、Transformation Designer の日付フォーマッタの使用が必要になる場合があります。たとえば、*January, 13, 2016*などの日付をフォーマット設定しようとする、TotalAgility の日付フォーマッタはフォーマット設定に失敗します。

一部のシナリオでは、Transformation Designer のフォーマット設定ルールを使用して、フォーマット設定と検証のために TotalAgility 抽出グループに渡される前に抽出値をクリアすることが効果的です。

Transformation Designer では、フォーマット設定と検証の有無にかかわらず、ドキュメントの抽出をテストできます。

- 抽出結果の改善に取り組んでいる場合は、フォーマット設定と検証を使用せずにテストすることが効果的です。
- ベンチマークの場合は、フォーマット設定と検証によるテストが重要になります。ベンチマークに使用されるゴールデン ファイルには通常、フォーマット設定と検証が行われます。これらのゴールデン ファイルのベンチマークを行うときは、フォーマット設定と検証を使用します。

フィールド フォーマッタ

TotalAgility Designer のフィールド フォーマッタ ([キャプチャ]>[フィールド フォーマッタ]) には、3 つの標準的なフォーマット設定方法である日付、金額、割合のほかに、1 つの非標準のフォーマット設定方法であるビジネス ルールが含まれます。

ビジネス ルールとは、開始ノードと終了ノードがあり、中間に 1 つ以上のアクティビティがある TotalAgility プロセスです。フォーマット設定ビジネス ルールには、必須の入力変数と出力変数がいくつかあります。これらの変数は大文字と小文字が区別され、ビジネス ルールが機能するように特定の順序で定義する必要があります。

必要な変数の構文:

- 入力変数 (名前、タイプ)
 - FieldText, string
- 出力変数
 - IsValid, boolean
 - ErrorMessage, string
 - FormattedText, string
 - FormattedValue, string

利用可能なアクティビティには多くのタイプがあります。式アクティビティを使用して、フィールド値をフォーマット設定し、Uppercase、Lowercase、Trim、Left、Mid、Right、Replace など、いくつかのセット変数を構成できます。

C# アクティビティと Visual Studio C# エディター

C# アクティビティは、ビジネスルールと比較すると、より優れたテスト機能とより優れた概要を提供します。

注 プロジェクト デザイナーは、このアクティビティに取り組むために Visual Studio C# の基本的な知識を身に付ける必要があります。

TotalAgility Designer の C# アクティビティの [構成] タブで C# コードを編集できます。[構成] タブのこのコード ボックスは小さく、構文の強調表示、IntelliSense、およびテスト機能はありません。Visual Studio の C# エディターを使用して、コードを作成、編集、およびテストすることをお勧めします。

C# アクティビティ コーディング ウィンドウで、開始に役立つ基本的なスケルトン コードを確認できます。このコードには ScriptParameter.sp パラメータが含まれます。同じパラメータを C# で使用する必要があります。これを行うには、C# で *Agility.Server.Scripting.dll* ライブラリを参照する必要があります。この DLL ファイルは通常、次の場所にあります。

- \\TotalAgility\CoreWorkerService\Agility.Server.Scripting.dll

例: 社会保障番号

米国の社会保障番号の形式は ddd-dd-dddd です。d は数字です。次の C# コードを使用して、社会保障番号を抽出するフィールドをフォーマットできます。

```
using System;
using Agility.Server.Scripting.ScriptAssembly;
namespace KTAformatter
{
    public class SocialSecurityNumber
    {
        [StartMethodAttribute()]
        public static void Method1(ScriptParameters sp)
        {
            string fieldText = sp.InputVariables["FieldText"].ToString();
            string formattedText = fieldText.Replace("-", "");
            formattedText = formattedText.Replace(" ", "");
            var num = 0;
            var isNumeric = Int32.TryParse(formattedText, out num);
            //check the length for 9 digits and the value to be numerical
            if (formattedText.Length == 9 && isNumeric)
            {
```

```
        formattedText = formattedText.Substring(0, 3) + "-" +
formattedText.Substring(3, 2) + "-" + formattedText.Substring(5, 4);
        sp.OutputVariables["FormattedText"] = formattedText;
        sp.OutputVariables["IsValid"] = true;
    }
    else
    {
        sp.OutputVariables["FormattedText"] = formattedText;
        sp.OutputVariables["ErrorMessage"] = "SSN Value is not a valid";
        sp.OutputVariables["IsValid"] = false;
    }
}
}
```

コードが正常に機能するとテストされたら、コードをコピーして、C# アクティビティ プロパティの [構成] タブのコード ボックスに貼り付けます。ビジネス ルールを保存し発出する前に、コードを検証してからテストします。

Validation

TotalAgility Designer には、抽出グループのクラス レベルに、次の標準タイプの検証メソッドがあります。日付、正規表現、および標準。これらのメソッドは、単一フィールドの検証ルールにのみ使用できます

「ビジネス ルール」は、単一フィールドの検証ルールに使用できる非標準の検証方法ですが、一般的には複数フィールドの検証ルールに使用します。

ビジネス ルールとは、開始ノードと終了ノードがあり、中間に 1 つ以上のアクティビティがある Kofax TotalAgility プロセスです。検証ビジネス ルールには、いくつかの必要な入力変数と出力変数があります。入力変数は、複数フィールドの検証ルールで使用されるフィールドです。単一のフィールドのみを使用している場合は、必要な変数は 1 つだけです。出力変数は 2 つあり、大文字と小文字が区別されません。ビジネス ルールが機能するためには次の順序で使用する必要があります。

- 入力変数 (名前、タイプ)
 - <FieldName_1>, string
 - <FieldName_2>, string
 - .
 - .
 - .
 - <FieldName_n>, string
- 出力変数
 - IsValid, boolean
 - ErrorMessage, string

ビジネス ルールがシンプルで管理可能なプロセスであり、多くのアクティビティを必要としない場合は、TotalAgility Designer の C# アクティビティの [構成] タブを使用します。たとえば、次のシナリオのようにスクロールまたは縮小することなくプロセスを 1 つの画面で表示できる場合は、C# アクティビティを使用します。Amount1 + Amount = Total。

これは、1 つの決定ノードと 2 つの式ノードで管理できます。

プロセスが複雑で、多くのアクティビティが必要な場合は、Visual Studio C# エディタを使用します。たとえば、プロセスでスクロールや縮小が必要な場合は、Visual Studio C# エディタを使用します。

アクティビティのロジックが非常に複雑で理解しにくい場合は、カスタム DLL を作成し、.NET アクティビティの Kofax TotalAgility ストアを介してその DLL を使用します。

例: 国際銀行口座番号 (IBAN)

ビジネス プロセスで IBAN を検証するためのロジックには多くのアクティビティが含まれるため、乱雑になりがちです。

C# アクティビティ コード ボックスには、開始するのに役立つ基本的なスケルトン コードが表示されます。このコードには、スクリプト パラメータ sp があります。同じパラメータを C# で使用する必要があります。そのためには、C# で Agility.Server.Scripting.dll ライブラリを参照します。この dll は次の場所にあります。\\TotalAgility\CoreWorkerService

次の C# コードは読みやすく、保守が簡単です。Visual Studio C# エディタで開発およびテストできます。

```
namespace StandardCaptureValidators
{
    using System;
    using Agility.Server.Scripting.ScriptAssembly;
    public class Iban
    {
        .net activity</param>
        [StartMethod]
        public void ValidateIban(ScriptParameters scriptParameters)
        {
            var iban = scriptParameters.InputVariables["Iban"].ToString();
            if (this.ValidateGermanIban(iban))
            {
                scriptParameters.OutputVariables["[IsValid]"] = true;
            }
            else
            {
                scriptParameters.OutputVariables["[ErrorMessage]"] = "IBAN is not valid";
                scriptParameters.OutputVariables["[IsValid]"] = false;
            }
        }
        /// <summary>Validate German Bank Account Number</summary>
        /// <param name="iban">The International Bank Account Number</param>
        /// <returns>True, if it is a valid German Iban number</returns>
        public bool ValidateGermanIban(string iban)
        {
            // D=13 & E=14 & 00 -> 131400
            var numericalCountryCode = "131400";
            var checkSum = iban.Substring(2, 2);
            var calculatedCheckSum = Convert.ToString(98 -
this.CalculateIbanCheckNumber(iban, numericalCountryCode));
            return checkSum.Equals(calculatedCheckSum);
        }
        /// <summary>Calculate the Internation Bank Account Number checksum</summary>
        /// <param name="iban">The International Bank Account Number</param>
        /// <param name="numericalCountryCode">The Country Code coming</param>
        /// <returns>The check number</returns>
        private int CalculateIbanCheckNumber(string iban, string numericalCountryCode)
        {
            var temp = iban.Substring(4) + numericalCountryCode;
            var part1 = Convert.ToString((Convert.ToInt32(temp.Substring(0, 6)) % 97));
```

```
var part2 = Convert.ToString((Convert.ToInt32((part1 + temp.Substring(6, 6))
% 97)));
var part3 = Convert.ToString((Convert.ToInt32((part2 + temp.Substring(12,
6))) % 97));
return Convert.ToInt32((part3 + temp.Substring(18))) % 97;
}
}
```

ビジネス ルールでのテーブル入力のキャプチャ

テーブル セルの検証に加えて、テーブルを検証できます。たとえば、空のテーブルに無効のマークを付けて、検証アクティビティが完了しないようにします。

検証ビジネス ルールでは、テーブル列とテーブル全体 (テーブル フィールド) の両方を入力として受け入れられます。

検証ルールが個々のテーブル列およびテーブル フィールド全体を対象としている場合は、1つの複数フィールド検証ビジネス ルール内にすべてのロジックを含めます。単一フィールドの検証ルールを個々のテーブル列に適用しないでください。また、同じテーブルが入力として使用される別の複数フィールド検証ビジネスルールを適用しないでください。

複数フィールド検証ビジネス ルールでテーブル フィールドが有効に設定されている場合、そのテーブル内のすべての行は有効としてマークされます。同様に、複数フィールドの検証ビジネス ルールでテーブルフィールドが無効に設定されている場合、そのテーブル内のすべての行は無効としてマークされます。そのため、セルはテーブル列の単一フィールド検証ルールにより無効とマークされていても、テーブルを有効に設定する複数フィールド検証ビジネス ルールが優先されます。この状態を回避するには、1つの複数フィールド検証ビジネス ルール内にテーブルのすべての検証ロジックを含めます。

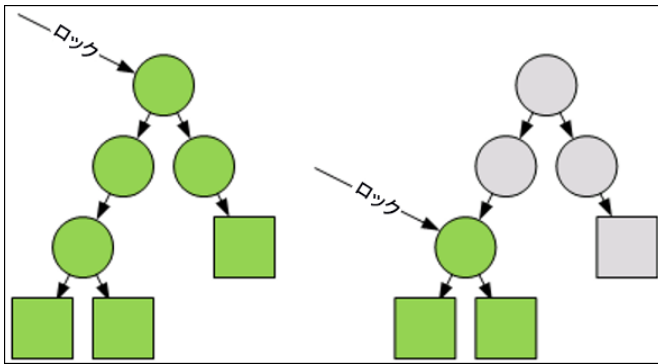
複数フィールド検証ビジネス ルールが入力としてテーブル フィールドを使用していない場合、または単一フィールド検証ルールがどのテーブル列にも割り当てられていない場合は、このルールに従わないでください。

第 11 章

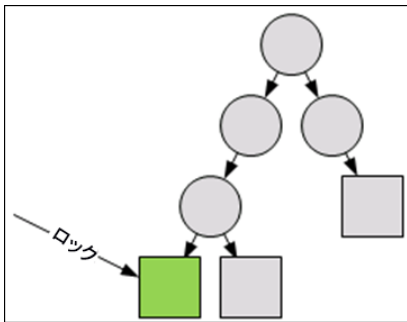
フォルダとドキュメントのロック

キャプチャプロセスを設計する際には、ドキュメントとフォルダのロック動作を理解することが重要です。ロックは、ドキュメントレベルまたはフォルダレベルで発生することが可能です。

フォルダの排他ロックにより、ユーザーはフォルダとドキュメントの両方を含むフォルダ階層内のオブジェクトを変更できます。ロック状態は階層の下方向に伝搬されます。つまり、フォルダをロックすると、そのフォルダ内のオブジェクトを変更できます。



ドキュメントの排他ロックは、ロックされたドキュメントの更新権限をユーザーに付与します。ただし、ドキュメントの削除、分割、または別のフォルダへの移動はできません。これを行うには、そのドキュメントが含まれているフォルダのロックを取得する必要があります。



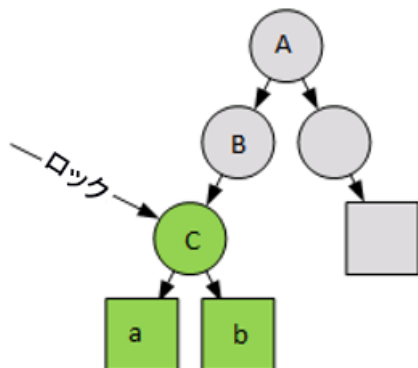
スキャン新規ジョブ作成 (SCNJ) は、フォームがロードされるとフォルダを作成し、ロックします。無人および有人のキャプチャ アクティビティにもロックが必要です。

次のシナリオでは、ロックの問題を回避するよう注意してください。

オブジェクトの削除

オブジェクトを削除するには、階層内でオブジェクトのすぐ上のフォルダの排他ロックが必要です。

次のロックシナリオでは、ドキュメント「a」または「b」は削除できますが、フォルダ「C」は削除できません。



フォルダ「C」を削除するには、フォルダ「B」の排他ロックが必要です。

サブフォルダで動作している複数のサブプロセスを並列して実行している場合は、削除アクティビティを追加する際のロック動作を考慮してください。削除アクティビティは、削除されるサブフォルダの親のロックを取得しようとします。

- 削除アクティビティがサブプロセスにある場合、サイズ1のスレッドプールに削除アクティビティを配置できます。これにより、一連の削除が強制され、ロックの競合が回避されます。
- または、削除アクティビティを親プロセスに配置し、すべてのサブプロセスが完了した後に親フォルダを削除します。

フォルダまたはドキュメントの入力変数

ドキュメントがキャプチャ アクティビティまたはプロセスへの入力として使用される場合。たとえば、部分完了機能は、分類アクティビティによって作成された各ドキュメントに対して新規ジョブを作成できます。作成されたジョブのキャプチャ アクティビティには、ドキュメントの排他ロックが必要です。ドキュメントを含むフォルダがロックされているときに、別のユーザーがそのドキュメントでキャプチャ アクティビティを実行しようとする、次のエラーが発生します。

親フォルダの1つがロックされているため、ドキュメントのロックを解除できません。技術的な支援が必要な場合は、アクションを参照してください: `DocumentReviewControlLoadBatchAction`。

または、ドキュメントがロックされていて、別のユーザーがそのフォルダでキャプチャ アクティビティを実行しようとする、次のエラーが発生します。

フォルダ階層にはすでにロックされたオブジェクトが含まれているため、フォルダをロックできません。技術的な支援が必要な場合は、アクションを参照してください: `ValidationControlLoadbatchAction`。

ループ ノード、またはドキュメントに直接作用するプロセス設計を使用すると、同様のロックの問題が発生する可能性があります。

ロックの衝突を回避するには、プロセス設計に競合を回避するためのロジックが必要になります。これを達成するための 1 つの方法は、事前条件を使用して、潜在的に競合するジョブまたはアクティビティが完了するまでプロセスの実行を強制的に待機させることです。

Web Capture コントロール

Web Capture コントロールを使用して臨時キャプチャ フォームを作成するときは、注意してロック動作を管理してください。Web Capture コントロールは、スキャンまたはアップロードの開始時に使用されるフォルダをロックします。ユーザーがツールバーの [保存] ボタンをクリックすると、ロックが解除されます。

ユーザーに [保存] ボタンを表示しないことを選択している場合は、Web Capture コントロールで保存アクションを呼び出してフォルダのロックを解除することもできます。この操作は、フォームのユーザーが Web Capture コントロールの外部のフォルダ コンテンツを操作できるようにする前、および後続のキャプチャ アクティビティの前に行う必要があります。

第 12 章

キャプチャ クライアント イメージ ビューアの画質

イメージ ビューアで画像を非ネイティブ解像度で表示するときに、TotalAgility は各ブラウザに組み込まれたイメージ スケーリング アルゴリズムを使用してイメージ スケーリングを実行します。そのため、ユーザーに表示されるイメージの品質はブラウザごとに異なる場合があります。TotalAgility によるスケーリング タスクの実行は各ブラウザに依存するため、このスケーリングによる画質への影響は TotalAgility によるコントロールの範囲外になります。

Internet Explorer 9 以降では、画像の縮小率があまり良くないため、サイズ縮小時に画質が低下することが知られています。Chrome ブラウザでは、画像のサイズ縮小がより適切に機能します。

イメージの表示品質が作業プロセスにとって重要な場合は、Internet Explorer ではなく Chrome ブラウザを使用します。

第 13 章

オンライン学習

オンライン学習は、失敗した分類または抽出結果を使用して、将来処理されるドキュメントを改善する方法です。ドキュメントが正常に分類されず、その後の本番稼働時に正しく分類された場合、そのドキュメントをトレーニングすることで以降の類似ドキュメントを正常に分類できるようになります。抽出結果を修正することでも、プロジェクトに同様の利点をもたらします。

オンライン学習システム タスク

オンライン学習は TotalAgility のシステム タスク「オンライン学習」により実行されます。このタスクは、ユーザーまたはシステムによりオンライン学習用としてフラグ付けされたすべてのドキュメントを取得します。システム タスクは、新しいドキュメントを含む動的ナレッジ ベースの新しい、修正されたバージョンをコンパイルします。タスクが完了すると、新しい、改善されたナレッジ ベースが Transformation Server で利用可能になり、処理される次のジョブで使用できます。

システム タスクが実行されるたびに、新しいナレッジ ベース ファイルが作成されます。このファイルは大きくありませんが、古いファイルは、それらを必要とするジョブが残っている可能性があるため、すぐには削除されません。

次の 2 つのベスト プラクティスを検討してください。

- オンライン学習システム タスクは、実行回数が 1 時間に 1 回までになるように構成します。請求書シナリオなどでは、通常は 1 日に 1 回でも十分です。このようなタスクは毎分実行しないでください。新しいナレッジ ベース ファイルが作成されますが、効果が顕著になるのは、学習が行われたタイプのドキュメントが処理されたときのみです。請求書の場合は、新しいベンダーが同じ日に 2 つの請求書を送信する可能性は低いいため、処理は翌日までに発生しません。たとえ発生したとしても、オペレータがこれらの 2 つの請求書を手動入力するのはそれほど負担ではありません。
- ナレッジ ベース ファイルは、TotalAgility のインストール場所に保存されます。通常は C ドライブです。オンライン学習システム タスクを 1 日に複数回実行する場合は、C ドライブに十分なスペースを確保してください。

トレーニング ドキュメントをインポートする間隔

プロジェクトが最初に本番稼働するときは、分類と抽出の結果は理想的な状態ではありません。プロジェクトが最初に本番稼働するときに、ほとんどのトレーニング ドキュメントが蓄積されますが、その後徐々に減少します。

抽出トレーニング ドキュメントが蓄積されると、それらは動的抽出ナレッジ ベースに置かれますが、他の抽出方法が失敗した場合にのみアクセスされます。これは次の 2 つのことを意味します。

1. このナレッジ ベースのドキュメントは、他のすべてのオプションが失敗するまで使用されません。これは、抽出パフォーマンスが低下する可能性を示唆しています。
2. 動的抽出ナレッジ ベースのサイズが大きくなると、プロジェクトのパフォーマンスが低下する可能性があります。

動的抽出ナレッジ ベースが大きくなりすぎることを防ぎ、抽出トレーニング ドキュメントを定期的に Transformation Designer 抽出セットに移動するには、新しい抽出サンプルをインポートし、競合を解決して、プロジェクトを定期的にトレーニングします。

分類トレーニング ドキュメントが蓄積されると、それらは動的分類子によって使用されます。抽出と同様、トレーニング ドキュメントの数が増えると、Transformation Server のパフォーマンスが低下する可能性があります。分類トレーニング ドキュメントと抽出トレーニング ドキュメントの両方を Transformation Designer にインポートし、分類と抽出の両方についてプロジェクトを定期的にトレーニングすることをお勧めします。

注 収集されたドキュメント数および [新規サンプル] ドキュメント セットにコピーされたドキュメント数は、Transformation Designer を使用していつでも確認できます。

間隔について次の例を検討してください。

- 1 週間後
- 2 週間後
- 3 週間後
- 4 週間後
- 2 か月後
- 3 か月後
- 6 か月後
- 1 年後

処理するドキュメントの量と蓄積されるトレーニング数に応じて、間隔を設定または変更します。

1 年後に、プロジェクトがドキュメントを正常に処理している必要があります。トレーニング ドキュメントは、新規ベンダーまたはフォームが発生したときにのみ収集されます。引き続きプロジェクトの監視とドキュメントのインポートを行い、半年ごとにプロジェクトを再トレーニングします。

第 14 章

パッケージの自動エクスポートおよび自動インポート

パッケージの自動エクスポートには、次の SDK API を使用します。

1. `PackageService - ExportPackageToBytes()`: この API は、パッケージを TotalAgility サーバーから呼び出し元に渡されたバイト配列にエクスポートします。

注 これが推奨されるアプローチです。

2. `PackageService - ExportPackageToFile()`: この API は、パッケージを 1 つのファイルにエクスポートします。このファイルは、TotalAgility サーバーからアクセスできる必要があります。

パッケージの自動インポートには、次の SDK API を使用します。

1. `PackageService - ExportPackageFromBytes()`: この API は、呼び出し元から TotalAgility サーバーに渡されたバイト配列からパッケージをインポートします。

注 これが推奨されるアプローチです。

2. `PackageService - ExportPackageToFile()`: この API は、1 つのファイルからパッケージをインポートします。このファイルは、TotalAgility サーバーからアクセスできる必要があります。

第 15 章

TotalAgility アプリケーションのパフォーマンスの監視

TotalAgility は、多数の重要なサービスの主要インジケータを測定するための Windows パフォーマンス カウンタが含まれています。詳細については、『Kofax TotalAgility 管理ガイド』の「パフォーマンス カウンタの使用法」を参照してください。

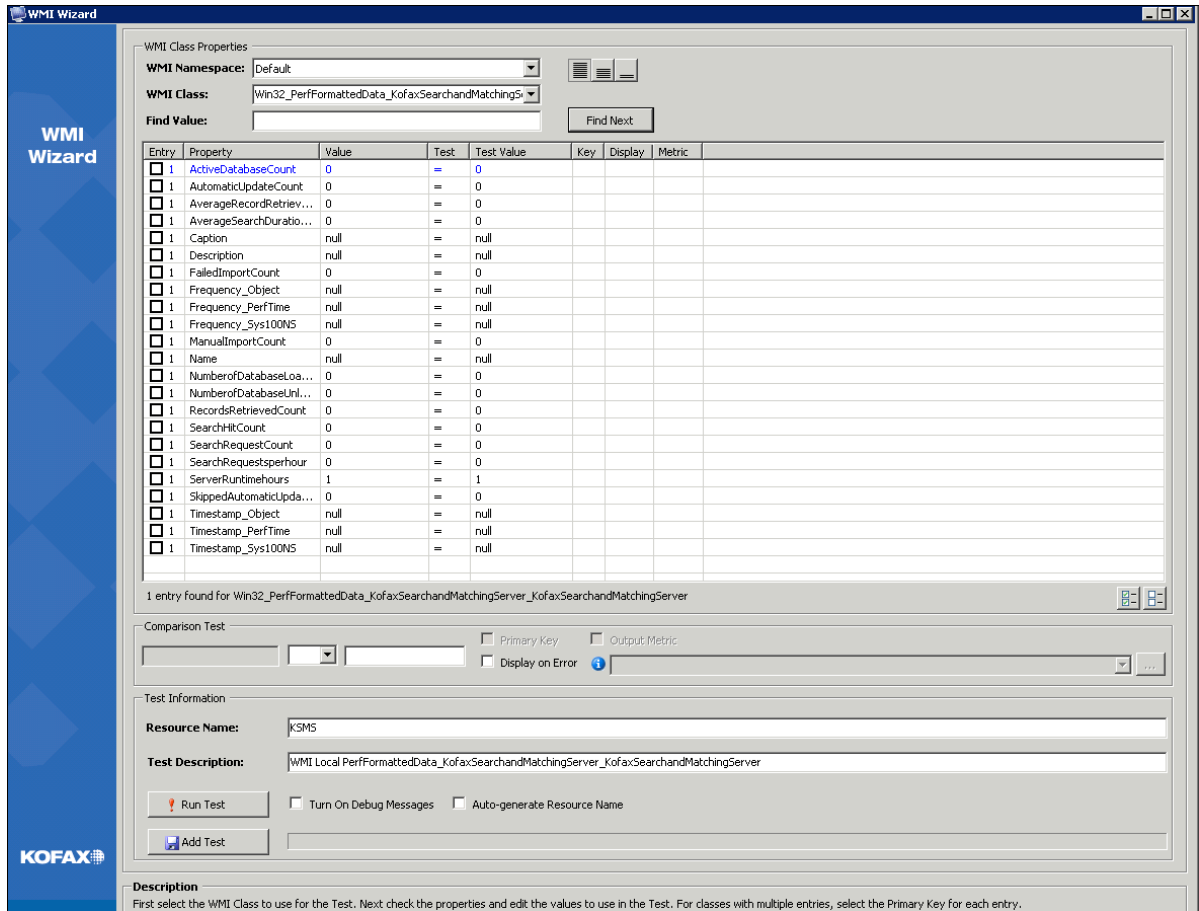
Kofax Monitor を使用してこれらのカウンタを監視し、値が定義された範囲外にある場合にアラートを送信できます。

TotalAgility Windows パフォーマンス カウンタの監視を有効にするには、次の手順を実行します。

1. WMI Performance Adapter Windows サービスがターゲット TotalAgility マシンで実行されていることを確認します。
2. Kofax Monitor サーバーから TotalAgility サーバーへの WMI 接続を確認します。「Microsoft wbemtest」ユーティリティを使用して、リモート WMI アクセスを確認します。wbemtest の詳細については、Microsoft のサイトを参照してください。
3. Kofax Monitor サーバーで、<km インストール パス>\ReveilleSoftware\Reveille\にある wmiclasses.ini ファイルの既存の名前空間セクション、[wmiclasses,NameSpace=\root\cimv2,Name=Default]:' に、次の WMI クラス定義を追加します。

```
wmiclass=
  Win32_PerfFormattedData_KofaxSearchandMatchingServer_KofaxSearchandMatchingServer
wmiclass= Win32_PerfFormattedData_KTACHildExportProcesses_KTACHildExportProcesses
wmiclass= Win32_PerfFormattedData_KTACoreWorkerLockedActivities_
  KTACoreWorkerLockedActivities
wmiclass=
  Win32_PerfFormattedData_KTACoreWorkerSystemTasks_KTACoreWorkerSystemTasks
wmiclass= Win32_PerfFormattedData_KTACoreWorkerThreads_KTACoreWorkerThreads
wmiclass=
  Win32_PerfFormattedData_KTACoreWorkerWorkerTasks_KTACoreWorkerWorkerTasks
```


4. Kofax Monitor 管理モジュールで WMI ウィザードを実行し、次のような WMI クラスを選択します。Win32_PerfFormattedData_KofaxSearchandMatchingServer_KofaxSearchandMatchingServer。



5. 目的のプロパティの WMI テストを作成し、Kofax Monitor ダッシュボード表示用にオプションで結果をメトリックに保存します。
 - a. TotalAgility パフォーマンス メトリックを有効にし、必要な評価しきい値を設定します。

Metric - Kofax.KTA.CoreWorkerThreads

Metric Evaluations

Name: Kofax.KTA.CoreWorkerThreads

Description: KTA Core WorkerThreads

Cost 1: 0.000 Description:

Cost 2: 0.000 Description:

Enabled

Save Results

OK Cancel

- b. スケジュールを割り当て、新しい TotalAgility パフォーマンス モニターへの通知を有効にします。

KOFAX Administration / Schedules / Assignments

Edit Schedule Assignments for Monitor - KTA_Performance

Schedule: Everyday 24x7 Assigned To: Monitor KTA_Performance Result: OK

Settings Notifications Schedules

Assignment Type: Monitor

Monitor: KTA_Performance

Available Schedules

- Holidays
- WeekDays Only
- WeekEnds Only

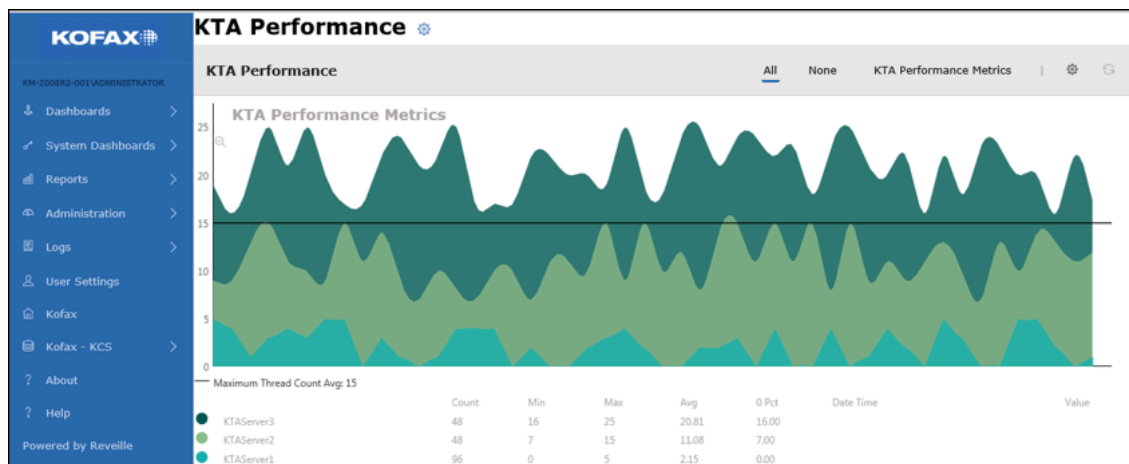
Assigned Schedules

- Everyday 24x7

Assign -->> <<-- Unassign

View Done

- c. TotalAgility パフォーマンス メトリックを表示する Kofax Monitor ダッシュボードを作成します。



注 Kofax Monitor で設定なしで使える TotalAgility 向けのサポートについては、Kofax Monitor ドキュメントに含まれる『Kofax Monitor ウィザードの使用』ドキュメントを参照してください。

第 16 章

TotalAgility リンク済みサーバーの監視

TotalAgility リンク済みサーバーの構成は、システム間でドキュメントを移動するために、2つの異なる TotalAgility インストール間に双方向リンクを定義します。詳細については、Kofax TotalAgility のドキュメントを参照してください。

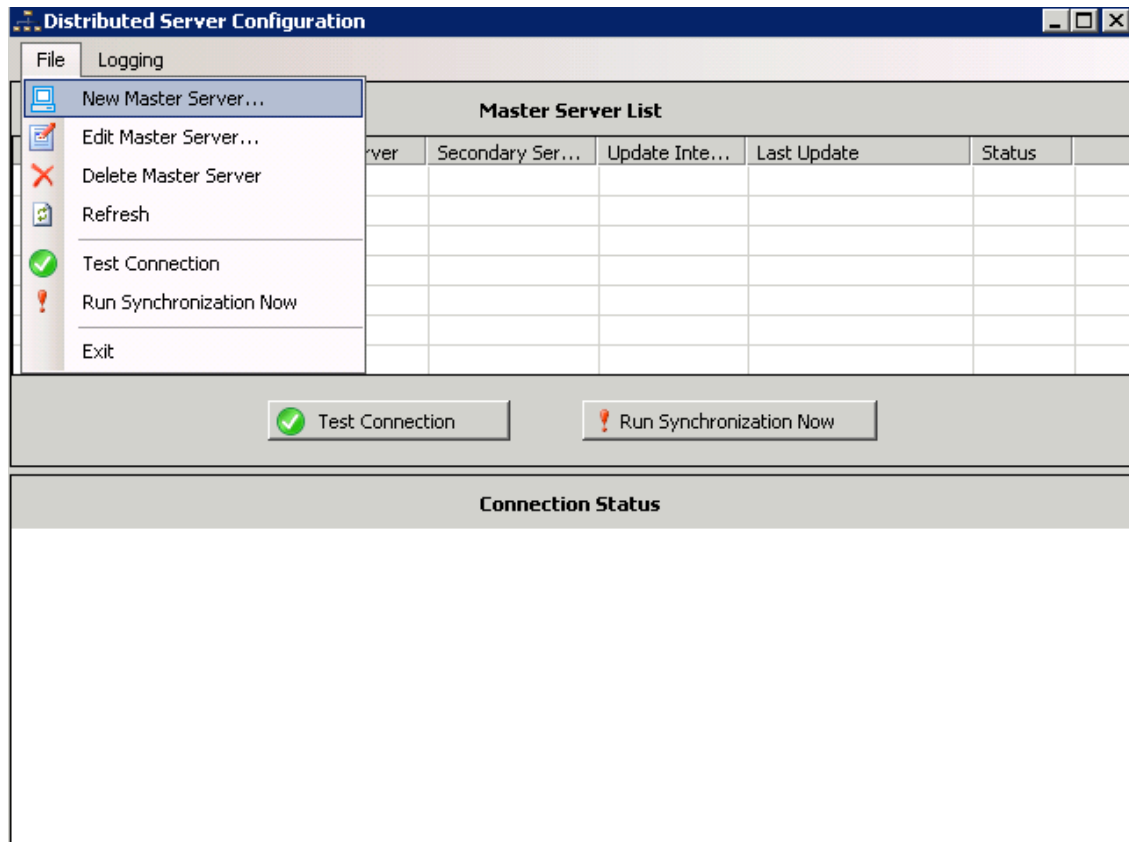
Kofax Monitor の分散サーバー機能により、Kofax Monitor サーバーは複数の TotalAgility リンク済みサーバーの場所でプロアクティブに動作し、各場所で TotalAgility Monitor の同一セットまたは別のセットを実行できます。Kofax Monitor リモートサーバーは、ローカルの TotalAgility Monitor を実行し、暗号化された Web サービス接続を介して監視結果を送信します。さらに、オプションでローカルアラートを Kofax Monitor マスターサーバーに 1 つ以上送信できます。

各モニターが Kofax Monitor マスターサーバー上に配置され、実行されているかのように、Kofax Monitor Remote TotalAgility Monitor のステータスとレポートを表示できます。

TotalAgility リンク済みサーバーで Kofax Monitor の分散サーバー機能を有効にするには、次の手順を実行します。

1. Kofax Monitor をそれぞれの TotalAgility リンク済みサーバーの場所にインストールします。
『Kofax Monitor インストールガイド』を参照してください。
2. TotalAgility ウィザードを使用して、それぞれの TotalAgility リンク済みサーバーの場所で TotalAgility を作成します。
『Using Kofax Monitor Wizards Guide』(Kofax Monitor ウィザードの使用ガイド)の「Using the Kofax TotalAgility Wizard」(Kofax TotalAgility ウィザードの使用)セクションを参照してください。

3. Kofax Monitor 管理コンソールで Kofax Monitor 分散サーバーを構成して Kofax Monitor 分散サーバー接続を作成します。Kofax Monitor 分散サーバー接続は、オプションに SSL 暗号化を有効にした Web サービスを使用して、Kofax Monitor 分散サーバー間で通信します。
 - a. Kofax Monitor リモート サーバーで、**[Kofax Monitor Admin Console]** を開始し、**[ファイル]** > **[分散サーバーの構成]** を選択します。



- b. **[ファイル]** > **[新しいマスター サーバー]** を選択します。

Master Server Configuration

Enabled

Local Server ID:
KM-2008R2-001

Primary Master

Server Address: (Example: localhost, localhost:8888)
KM-Master-01

Use SSL

SSL Client Certificate: [] ...

Secondary Master

Server Address: (Example: localhost, localhost:8888)

Use SSL

SSL Client Certificate: [] ...

Credentials
KMAAdmin [] ...

Update Interval
15 [] Minutes (+-5)

Save Cancel

- c. Kofax Monitor マスター サーバー接続を定義し、[保存] をクリックします。
- d. マスター サーバーの構成を保存します。
接続は、Kofax Monitor マスター サーバーで明示的に有効にされるまで無効のままです。
- e. Kofax Monitor リモート サーバー接続を有効にするには、次の手順を実行します。

Administration / Remote Servers Update Threshold Target Minutes Min: Max:

Remote Server: KM-2008R2-001 Enabled

Enabled	Reset	Delete	Remote Server	Last Update	Status	Duration
<input checked="" type="checkbox"/>	⏪	✕	KM-2008R2-001	[Reset]	Remote Server Disabled at the Master Server	

Successful sync in last 30 minutes No successful sync in last 30 minutes No successful sync in last 60 minutes

Maximum number of days to send on initial synchronization:

1. Kofax Monitor ユーザー コンソールの [管理] > [リモート サーバー] を選択します。
 2. 接続を有効にします。
 3. [保存] をクリックします。
- f. Kofax Monitor リモート サーバーからの接続を確認するには、[分散サーバーの構成] で [接続のテスト] をクリックします。