ホワイトペーパー

Solutions Business Manager の企業における本番環境までのパス

opentext™

目次

ページ

Business Manager Composer では、使い慣れたドラッグアンドドロップインターフェイスを使用してワークフローやフォームなどのプロセスアーティファクトを視覚的に設計できます。

Solutions Business Manager の開発プロセスの概要

OpenText™ Solutions Business Manager は、ビジネスプロセスの導入と開発を簡素化するためのものです。ワークフローやフォームなどのプロセスアーティファクトを、使い慣れたドラッグアンドドロップインターフェイスを使用して視覚的に設計できます。プロセスアプリの試用段階になったら、ボタンをクリックするだけでアプリを検証し、テスト環境に導入できます。また、環境に備わっている Application Administrator を使用して、ユーザー、グループ、プロジェクトをステージング環境に設定できます。導入準備が整ったら、Application Repository を使用すると、構成したプロセスをステージング環境から本番環境に簡単にプロモートできます。このホワイトペーパーでは、本番環境に導入するための標準パスと、代替方法としてのサンドボックスアプローチについて説明します。

本番環境までのパスに登場するもの

本番環境までのパスとは、設計を本番環境に導入し、エンドユーザーが利用可能な状態になるまでに通る必要のある各段階を示します。Solutions Business Manager では、Solutions Business Manager Composer、Solutions Business Manager Application Repository、さまざまな Solutions Business Manager ランタイム環境が相互に作用します。これらのコンポーネントは次のような役割を果たします。

Solutions Business Manager Composer

Solutions Business Manager Composer は、プロセスアプリの設計で使用するクライアントアプリケーションです。Composer では、アイテムを表すために収集するデータ、組織内でのアイテムの移動と更新の方法、組織内での役割、ユーザーがアイテムデータの表示や変更で使用するフォームを定義します。

Solutions Business Manager Application Repository

Solutions Business Manager Application Repository には 2 つの役割があります。1 つが、設計の個々のコンポーネント (フォーム、ワークフローなど) のソース管理リポジトリとしての役割。もう 1 つが、このホワイトペーパーで重視される、プロセスアプリの環境への導入と環境間での移動を管理するアプリケーションとしての役割です [1]。Application Repository は、公開される各バージョンのプロセスアプリを保管し、ランタイム環境とのすべての重要なインタラクションの記録を維持します。Solutions Business Manager の本番環境までのパスにおいて、中心的存在と言えます。

1 開発者はプロセスアプリを Business Manager Composer から社内の環境に直接導入できますが、その場合はまず、プロセスアプリを Application Repository に公開してから、対象の環境への導入を開始します。

環境の構築

その他に本番環境までのパスに登場するものとして挙げられるのが、**環境**です。これは、管理者とエンドユーザーのためのランタイムサービス一式を意味します。たとえば、Composer で設計されたフォームをエンドユーザーに提示するアプリケーションエンジン、自動 BPEL プロセスを実行して Web サービスを呼び出すオーケストレーションエンジン、関心を持つユーザーに通知を送信する通知サーバー、受信メールを処理するメールクライアントなどのランタイムサービスが含まれます。

Solutions Business Manager のインストールには、通常、複数の環境が含まれます。**開発環境**では、エンドユーザーに影響を与えずに変更内容をテストします。**ステージング環境**では、本番環境で使用する構成を複製して、構成したプロセスアプリをテストします。**本番環境**では、作成したプロセスアプリやソリューションを導入して、Solutions Business Manager のビジネスユーザーに価値を提供します。

どの環境にも Solutions Business Manager Application Administrator が含まれ、管理者がプロセスアプリのランタイムアスペクトを構成および調整できるようになっています。一般には、実行システムのうち、どの環境でも変わらない部分 (設計) を Composer で作成し、環境によって変化するアスペクトを Application Administrator で構成します。たとえば、ユーザー、グループ、プロジェクト、通知はすべて、管理対象であり、Application Administrator で構成します。また、プロセスアプリの動作をプロジェクトによって変更できるように設計を無効にする際も Application Administrator を使用します。

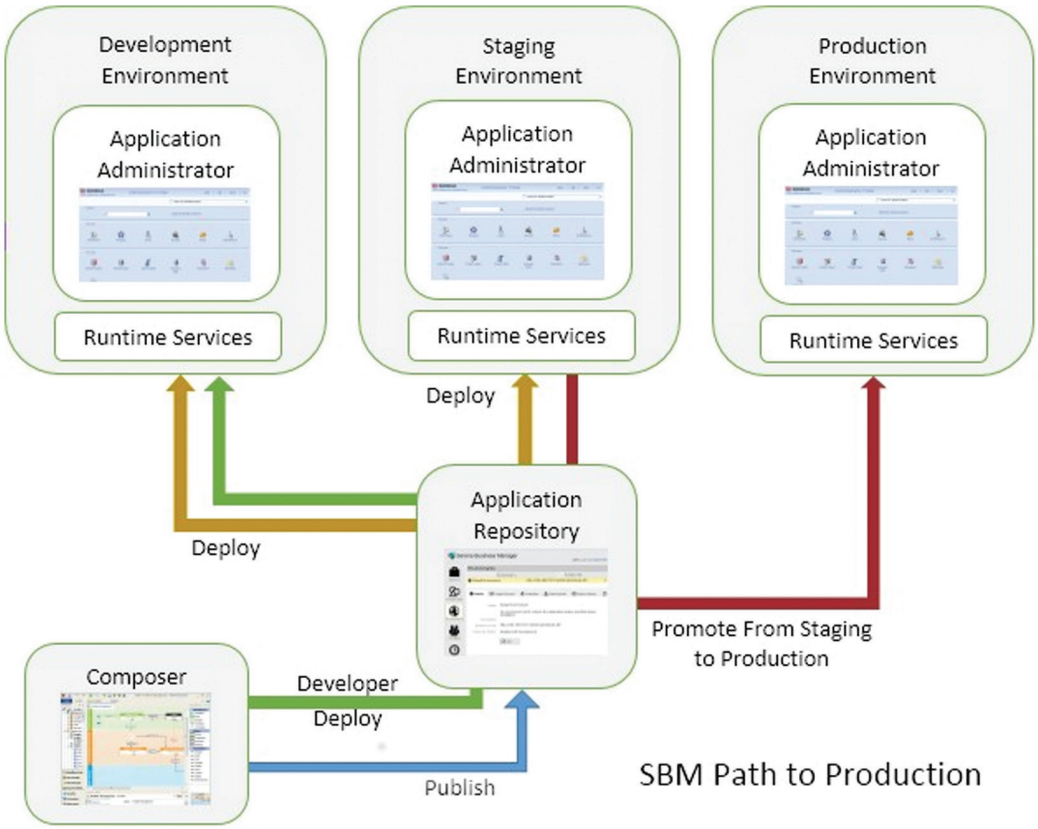
Solutions Business Manager のインストールには、通常、複数の環境が含まれます。

標準インストールトポロジ

Solutions Business Manager は、お客様の多様なニーズに対応するため、さまざまなトポロジでインストールできます。標準インストールには、開発、ステージング、本番の 3 種類の環境があり、1 つの Application Repository を使用して、プロセスアプリの環境への導入と環境間のプロモーションを仲介します。

次の図は、標準インストールにおける Composer、Application Repository、ランタイム環境の間の関係を示しています。

ビジネス開発者が、プロセスの作成または変更を開始します。開発中、プロセスアプリを調整し、開発環境に導入してテストします。この工程を、設計がステージング可能になるまで繰り返します。



本番環境までのパスのライフサイクル

次に、標準インストールでの本番環境までのパスの一般的なライフサイクルを説明します。

開発者が、Composer を使用してプロセスアプリの作成または変更を開始します。開発中、プロセスアプリを調整し、開発環境に導入してテストします。この工程を、設計がステージング可能になるまで繰り返します。開発を促進するため、管理者は開発環境のバージョン管理を停止することもできます [2]。

設計が完了したら、開発者がプロセスアプリを Application Repository に公開し、管理者がそのプロセスアプリをステージング環境に導入します。

2 Application Repository の環境には Composer プロパティがあり、これを［Enable Deployment］に設定すると、Composer から直接導入できるようになります。このプロパティを［Enable Development Deployment］に設定すると、Composer のユーザーが、変更内容をチェックインしたり、導入した設計図のバージョンを Application Repository で作成したりせずに、導入できます。

ステージング環境では、プロセスアプリが Application Administrator とレポーティングインターフェイスを使用して構成され、本番環境に必要なランタイム機能と管理機能を備えるようになります。たとえば、ユーザーが定義され、適切な権限と役割が付与されます。また、プロジェクトが定義されたり、プロジェクトの設計が上書きされることもあります。プロセスアプリの構成が完了したら、この環境で承認テストを実施し、本番環境でユーザーが使用できる状態であることを確認します。

最終的に、プロセスアプリが本番用として承認されると、Application Repository の**プロモート**機能を使用してステージング環境から本番環境にプロモートされます。プロモーションは導入とは異なります。設計では Solutions Business Manager Composer で作成された設計のみが移動するのに対し、プロモーションでは、基盤となる設計のほかにランタイム設定と管理設定が対象の環境に移動するためです。

開発プロシージャは企業によって異なるため、使用する Solutions Business Manager の構成もそれに合わせて変更する必要があります。

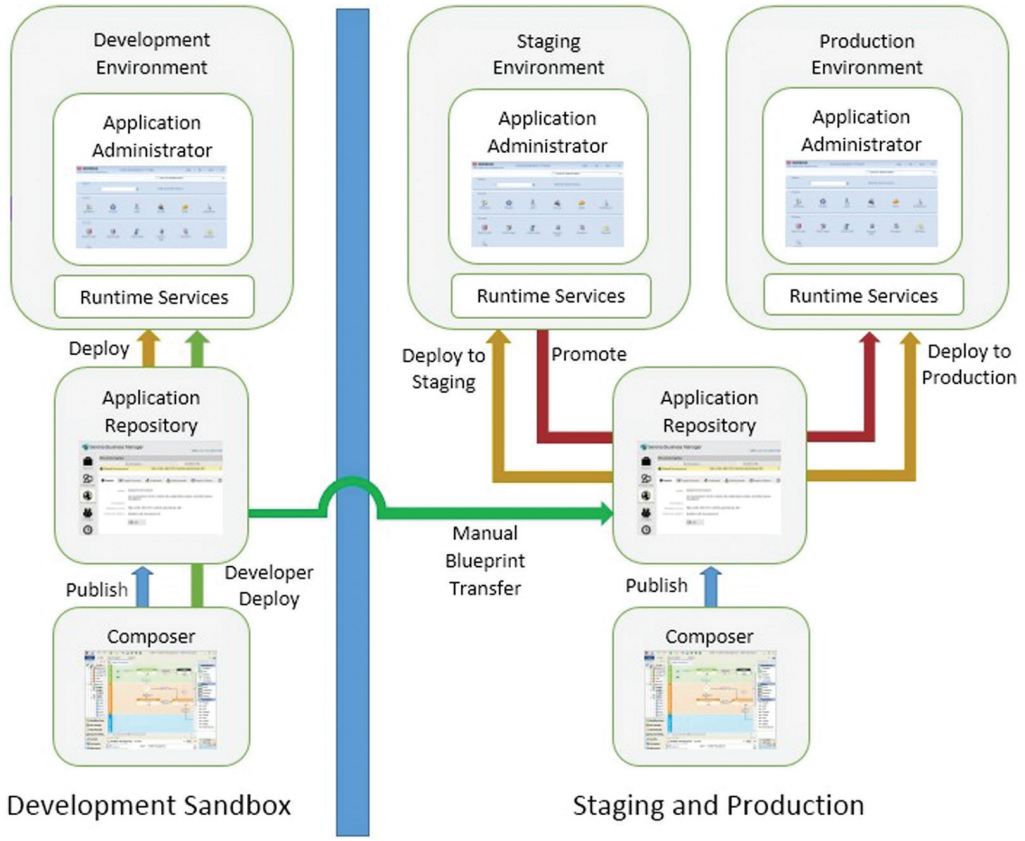
サンドボックス開発システムの使用

開発プロシージャは企業によって異なるため、使用する Solutions Business Manager の構成もそれに合わせて変更する必要があります。このセクションでは、サンドボックス開発システムとステージングおよび本番システムが物理的に分離していることが要件となる場合の設定に適した本番環境までのパスを説明します。このようなタイプの構成には、次のようなメリットがあります。

- 開発と本番の間でユーザーモデルや権限モデルの管理が容易
- 物理的に分離しているため、本番環境を物理的なセキュリティで保護可能
- 接続されていない開発作業をサポート

次の図は、サンドボックスシステムでの Composer、Application Repository、ランタイム環境の関係を示しています。

完全な管理特権を付与された開発者が、Application Repository や開発環境で必要な作業を実行できます。本番システムに不要な変更を加える危険性はありません。実際のところ、開発者はステージングシステムや本番システムでユーザーである必要はないのです。



独立した開発サンドボックスシステムを使用している場合、システムのユーザーはステージング環境にも本番環境にもアクセスできません。完全な管理特権を付与された開発者が、Application Repository や開発環境で必要な作業を実行できます。本番システムに不要な変更を加える危険性はありません。実際のところ、開発者はステージングシステムや本番システムでユーザーである必要はないのです。

このトポロジの場合、プロセスアプリの開発者は完全に開発サンドボックスだけで作業します。設計が完了し、ステージングに移る準備が整ったら、プロセスアプリが設計図ファイル (.msd) として Composer または Application Repository からエクスポートされ、ステージングおよび本番システムの Application Repository にインポートされます。これは、手動でプロセスアプリをサンドボックスシステムからステージングおよび本番システムの Application Repository に公開するのと同じ作業です。

また、サンドボックスでプロセスアプリに管理上の変更を加え、その変更を、Application Repositoryの手動プロモーション機能を使用してステージングシステムに移すこともできます。それには、管理上の変更を加えた後、開発環境からスナップショットを取得し、スナップショットファイル (.mss) としてサンドボックスの Application Repository からエクスポートします。そして、ステージングおよび本番システムの Application Repository にインポートし、そこからプロモーションを完了します。

その後、管理者がプロセスアプリをステージング環境に導入し、Application Administrator とレポーティングインターフェイスを使用する場合と同じようにステージングを続行します。

最終的に、プロセスアプリが構成され、本番用に準備が整ったら、その構成したプロセスアプリを管理者が Application Repository を使用して本番システムにプロモートします。

プロセスアプリが構成され、本番用に準備が整ったら、その構成したプロセスアプリを管理者が Application Repository を使用して本番システムにプロモートします。

本番環境でのプロセスアプリのパッチ適用

これまで説明してきたプロセスでは、ステージングおよび本番システムでプロセスアプリの設計を変更することはありません。つまり、設計の「信頼できる情報源」は、常に、接続されている開発システムと Application Repository なのです。これにより、開発プロセスはシンプルになりますが、実際にはこのプロセスを実行できない場合もあります。代替アプローチを検討する際は、次の点を考慮してください。

- 現時点で本番環境にあるプロセスアプリのバージョンを変更するのか？
- 本番環境向けの変更のみを適用するのか？
- プロセスアプリの現在開発中のバージョンにも同じ変更を適用しているか？
- 本番に追加しているID付きアイテムについて、同一アイテム(IDが同じ)をプロセスアプリの開発バージョンに追加したことを確認したか？

このセクションでは、ステージングシステムまたは本番システムに直接変更を加える際の問題について説明し、その問題に対応するためのアプローチを紹介します。

本番プロセスアプリに直接変更を加えるには、本番環境に導入されたバージョンのプロセスアプリのアクセス権を取得し、そのバージョンの設計に変更を加え、本番環境に再導入する必要があります。

本番環境にパッチを実装するための要件

本番プロセスアプリに直接変更を加えるには、本番環境に導入されたバージョンのプロセスアプリのアクセス権を取得し、そのバージョンの設計に変更を加え、本番環境に再導入する必要があります。開発作業がまだ完了していない場合、このような変更が誤って本番環境に適用されないようにし、希望のパッチ変更のみが適用されるようにする必要があります。また、本番環境に加えた変更を進行中の開発にも確実に反映し、今後、現在開発中の変更を導入する際にその変更が無効にならないようにする必要もあります。

システムにパッチを適用する際の問題は、状態などのアーティファクトを本番環境に加えるときに発生する場合があります。このようなアーティファクトは、名前ではなく固有の ID に基づいてシステムで特定されます。開発システムに同じ名前の状態を独自に追加しても、以前に本番環境に追加した状態と同じ ID にはなりません。つまり、後でそのプロセスアプリを本番環境に導入すると、同じ名前の状態が 2 つになり、一方はワークフローで使用されなくなるということです。この問題の回避方法は、標準インストールとサンドボックス開発システムを使用したインストールで異なります。

標準インストールでパッチを適用する際のアプローチ

単一の Application Repository を使用している (つまり、サンドボックスシステムを使用していない) 場合は、Composer の**パッチコンテキスト**機能を使用して、本番環境に導入されている設計に変更を加えることができます。Composer のパッチコンテキストでプロセスアプリを開くと、ラベル付きバージョンのプロセスアプリで作業することになります。この状態で、設計要素をチェックアウトし、変更を加え、チェックインし直します。続いて、変更したバージョンのプロセスアプリを公開し、導入します。この作業は、プロセスアプリの新たな開発が進行中でも、その中断を最小限に抑えて実行できます。

パッチコンテキストでの変更には、後で ID の衝突が発生しないように、制約がいくつかあります。たとえば、新しい状態やフィールドを直接追加できません。ただし、開発情報から状態やフィールドをコピーすることはできます。これにより、本番システムと開発情報の間でアイテムの相違を回避できます。そのため、パッチにフィールドを追加したい場合は、まず開発システムにそのフィールドを追加し、続いてテーブルをパッチコンテキストでチェックアウトし、テーブルの **[Add Existing Field...]** コンテキストメニュー項目を使用してフィールドをパッチコンテキストにコピーする必要があります。必要な変更を加えてパッチコンテキストを更新したら、パッチを適用したバージョンのプロセスアプリを公開して、ステージング環境や本番環境に導入できます。

パッチコンテキストを開く

［File］メニューの［Open...］を選択して、［Open Process App］ダイアログを起動します。［Look in: Repository］ラジオボタンを選択して、パッチを適用するプロセスアプリを選択します。［Open labelled version...］ボタンをクリックします。ダイアログに公開済みバージョンの一覧が表示されます。本番環境にあるバージョンを選択して、［OK］をクリックします。これまでにこのバージョンをパッチコンテキストとして開いたことがない場合は、選択したバージョンラベルに基づくパッチコンテキストの作成を促すダイアログが表示されます。[Yes]をクリックしてパッチコンテキストを作成します。パッチコンテキストで Composer を使用している場合は、タイトルバーのプロセスアプリ名とバージョンラベルの後に「[Patch]」が表示されます。

Solutions Business Manager では ID の衝突を回避できません。代わりに、今後、開発中のバージョンを導入する際に、本番環境に加えたすべての変更が反映されていることを確認する必要があります。

標準インストールでパッチを適用する際のアプローチ

開発システムが独立している本番環境にパッチを適用する場合にも、すでに説明した懸案事項が該当します。しかし、個々のシステムが独立しているため、Solutions Business Manager では ID の衝突を回避できません。代わりに、今後、開発中のバージョンを導入する際に、本番環境に加えたすべての変更が反映されていることを確認する必要があります。この問題に対応する方法はいくつかあります。

パッチコンテキストを使用する

現時点で本番環境に導入されているプロセスアプリのバージョンがサンドボックスの Application Repository にある場合、そのラベルに基づいてサンドボックスの Composer でパッチコンテキストを作成し、すでに説明した方法で変更を加え、設計を設計図としてエクスポートできます。この時点で、設計図を本番システムの Application Repository にインポートし、ステージング環境または本番環境に導入できます。すでに説明したように、パッチコンテキストに直接加えた変更で開発情報を更新する必要があります。

Composer の比較マージ機能を使用する

Composer のマージ機能を使用する方法は 2 つあります。どちらも、一方のシステムに加えた変更がもう一方に正しく反映されます。変更内容は、サンドボックスシステムから本番システム (またはその逆) にコピーできます。どちらの方法を選択するかは、本番システムに加えたい変更がサンドボックスシステムにすでに加えられているかどうかによって変わります。

変更内容をサンドボックスシステムから本番システムにマージする：本番システムの変更がサンドボックスシステムにすでに加えられている場合、変更内容を本番システムにマージします。それには、まず、［File］-> ［Export...］の順に選択して、サンドボックスシステムからプロセスアプリの設計図ファイル (.msd) を取得します。次に、本番環境の Application Repository に接続されている Composer イン

2 つの状態間の遷移を追加する場合、本番システムとサンドボックスシステムの両方に、個別に変更を加えることができます。

スタンスを開き、本番環境にあるプロセスアプリのバージョンを開きます。もともと、プロセスアプリを本番の Composer にインポートして公開し、導入している場合は、そのプロセスアプリが［Open Process App］ダイアログ (［Repository］オプション) に青色と金色のアイコン付きで表示されます。プロセスアプリを Application Repository にインポートして、そこから導入している場合は、赤色のアイコンが付いています。

本番バージョンのプロセスアプリを Composer で開いたら、**［File］-> ［Compare］-> ［With Local File...］**の順に選択して、サンドボックスバージョンのプロセスアプリを含む設計図と本番バージョンを比較します。その後、比較したアプリ (サンドボックスシステムからの設計図) にある、**［Copy to Open Process App］**コンテキストメニュー項目を使用して、アイテムを本番環境に複製します。この方法では、設計要素を示す固有 ID が維持されるため、後で衝突が発生することはありません。

変更内容を本番システムからサンドボックスシステムにマージする：先に本番システムに変更を加えたい場合は、変更を加えた後、すでに説明した方法を逆順に実行して、変更内容をサンドボックスシステムにマージできます。この場合、プロセスアプリの設計図ファイルは本番システムから取得します。それには、設計図を Composer からエクスポートして、Composer でサンドボックスバージョンのプロセスアプリを開き、開発システムにマージします。

小さな変更を同時に加える

ランタイム時に固有 ID 付きのアーティファクトを作成しない変更の場合は、2 つのシステムを同時に変更できます。たとえば、2 つの状態間の遷移を追加する場合、本番システムとサンドボックスシステムの両方に、個別に変更を加えることができます。ただし、次の設計要素はシステムに個別に追加できないので注意してください。

- 表
- フィールド
- 選択肢
- アプリケーションワークフロー
- 状態
- レポート定義
- 役割
- アプリケーション
- オーケストレーション

これらの要素を追加すると、サンドボックスシステムのプロセスアプリのバージョンを本番システムに導入したときに、衝突が発生する原因となります。すでに説明した比較機能を使用して、このような小さな変更が両方のシステムに正確に加えられたか確認することもできます。

まとめ

Solutions Business Manager では、すべての環境で 1 つの Application Repository インスタンスを共有する標準インストールを使用すると、管理可能なわかりやすいパスで本番機能を実現できます。サンドボックスシステムを使用して開発している場合は、設計の相違が発生する可能性が高くなります。そのため、本書で説明しているように、発生する可能性のある問題と変更のマージ方法に注意する必要があります [3]。

詳細情報はこちら：
www.microfocus.com/opentext

3 変更のマージの難しさを最小限に抑える方法の詳細については、Solutions Business Manager の開発のベストプラクティスに関する文書も参照してください。

お問い合わせ
OpenText の CEO、
Mark Barrenechea のブログ